

# 开发月刊

Development Monthly

2012年03月

总第012期



**“Java已死” 简史**

**Facebook版《愤怒的小鸟》为何选用Flash**

**Android编程方法大PK：NDK vs. RenderScript**

3 3月编程排行榜:JavaScript回归

### 专题报道 《WEB开发性能优化》

6 在ASP.NET网站使用HTML 5拖放

8 刷新网站：用Ajax构建更好单页面体验

11 性能优化：更快速加载你的JavaScript

### 技术热点 Techlogy hot

12 宅男程序员给老婆计算机课程之设计模式

13 宅男程序员给老婆计算机课程之模板引擎

16 软件开发基本原则之最佳项目

19 微软发布SQL Server 2012 RTM版本

20 有可能挑战Java优势的四种技术

21 “Java已死” 简史

22 3月Java热点热评

23 Facebook版《愤怒的小鸟》为何选用Flash

25 IT裁员为何总是测试工程师最受伤

26 国内主流开放平台生态系统建设情况调研报告

### 移动开发 Mobile Develope

27 三大移动Web框架哪个适合你

30 如何把交易型元数据编程移动应用

32 不破不立？移动行业的创造性破坏

34 Android编程方法大PK：NDK vs RenderScript

36 美女开发者独立开发二十多款应用月入上万

39 杂谈：迷失的苹果新款iPad

### 特别推荐 Best topic

41 51CTO技术沙龙视频专题《大话PHP设计模式》



# 3月编程语言排行榜:JavaScript回归

TIOBE 近日公布了 2012 年 3 月份的编程语言排行榜,本月的排行榜最大的看点莫过于 JavaScript 连升 2 名重新回到了第 8 名的位置,超越了 Perl 和 Python。JavaScript 自 1995 年诞生以来已过去了 17 个年头,如今现在的网站越来越多都依赖 JavaScript 开发,以及目前 HTML5 和 node.js 也受到了不少开发者的火热追捧。

TIOBE 近日公布了 2012 年 3 月份的编程语言排行榜,本月的排行榜最大的看点莫过于 JavaScript 连升 2 名重新回到了第 8 名的位置,超越了 Perl 和 Python。在 09 年 3 月到 12 月连续 10 月稳居第 8 的排名辉煌如今又重新回来了。JavaScript 自 1995 年诞生以来已过去了 17 个年头,如今现在的网站越来越多都依赖 JavaScript 开发,以及目前 HTML5 和 node.js 也受到了不少开发者的火热追捧。预计 JavaScript 的普及不久将会得到进一步的提升。JavaScript 从过去装饰性的一种脚本语言转变为主流的编程语言,人们用它来开发更大更复杂的程序,这就需要 Web 开发人员更深入的了解 JavaScript,了解它的过去、现在和未来。

在排行榜中还有另一个有趣的现象,与软件相关的巨头公司,如甲骨文,微软,苹果推出的语言都挤进了 10 名内,相比之下谷歌似乎无法达到。谷歌的 GO 语言在上个月的排名跌出 50 名,而其 Dart 语言在本月的排行仅在 78 位。

本月的的前两名依然是 Java 和 C,但值得注意的是 Java 在本月的份额为 17.110%,比去年同期下降了 2.60%。而 C 在本月的份额为 17.087%,比去年同期增长了 1.82%。编程语言的王者之争不久很可能发生改变。而在 2012 年 1 月编程

排行榜中, C++ 被 C# 追上,丢失了季军位置,同时又被 Objective-C 紧追其后,两面受敌。看来 C++ 要重新回到季军的位置恐怕不是件容易事。

下面是本期编程语言排行榜的其他排名数据和趋势走向

下面是前 20 名的编程语言排行

Position Mar 2012	Position Mar 2011	Delta in Position	Programming Language	Ratings Mar 2012	Delta Mar 2011	Status
1	1	=	Java	17.110%	-2.60%	A
2	2	=	C	17.087%	+1.82%	A
3	4	↑	C#	8.244%	+1.03%	A
4	3	↓	C++	8.047%	-0.71%	A
5	8	↑↑↑	Objective-C	7.737%	+4.22%	A
6	5	↓	PHP	5.555%	-1.01%	A
7	7	=	(Visual) Basic	4.369%	-0.34%	A
8	10	↑↑	JavaScript	3.386%	+1.52%	A
9	6	↓↓↓	Python	3.291%	-2.45%	A
10	9	↓	Perl	2.703%	+0.73%	A
11	13	↑↑	Delphi/Object Pascal	1.727%	+0.73%	A
12	30	↑↑↑↑↑↑↑↑	PL/SQL	1.418%	+1.01%	A
13	11	↓↓	Ruby	1.413%	-0.09%	A
14	23	↑↑↑↑↑↑↑↑	Transact-SQL	0.925%	+0.38%	A
15	15	=	Lisp	0.922%	-0.01%	A
16	22	↑↑↑↑↑	Visual Basic .NET	0.784%	+0.22%	A-
17	18	↑	Pascal	0.771%	+0.07%	A
18	32	↑↑↑↑↑↑↑↑	Logo	0.717%	+0.31%	A-
19	17	↓↓	Ada	0.633%	-0.09%	B
20	19	↓	NXT-G	0.604%	-0.04%	B

JavaScript 前身

1995 年前后,随着因特网的发展,网页开始流行起来。



### 3 月编程语言排行榜 :JavaScript 回归 II

一些公司开始尝试发布自己的网页浏览器。

最早的公司是称作 Nombas 的。它开发了一种叫 c 减减的程序。后因为名称消极更名为 ScriptEase。此脚本语言是被捆绑在 CEnvi 的共享软件产品中。该公司开发了嵌入网页的 CEnvi 版本。

1995 年 Netscape 和 sun 公司联手开发了一个叫 livescript 的脚本语言。就在 Netscape Navigator 2.0 即将正式发布前, Netscape 将其更名为 JavaScript, 目的是为了利用 Java 这个因特网时髦词汇。Netscape 的赌注最终得到回报, JavaScript 从此变成了因特网的必备组件。

1997 年 ECMA 统一标准 ECMAScript 脚本语言。

#### 三足鼎立

因为 JavaScript 1.0 如此成功, Netscape 在 Netscape Navigator 3.0 中发布了 1.1 版。恰巧那个时候, 微软决定进军浏览器, 发布了 IE 3.0 并搭载了一个 JavaScript 的克隆版, 叫做 JScript (这样命名是为了避免与 Netscape 潜在的许可纠纷)。微软步入 Web 浏览器领域的这重要一步虽然令其声名狼藉, 但也成为 JavaScript 语言发展过程中的重要一步。

在微软进入后, 有 3 种不同的 JavaScript 版本同时存在: Netscape Navigator 3.0 中的 JavaScript、IE 中的 JScript 以及 CEnvi 中的 ScriptEase。与 C 和其他编程语言不同的是, JavaScript 并没有一个标准来统一其语法或特

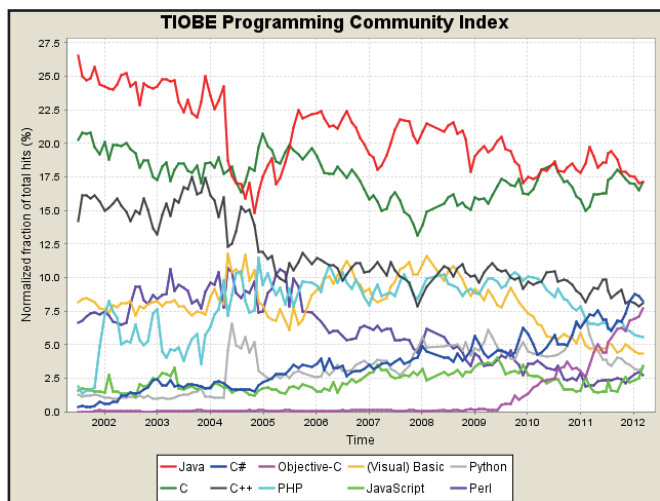
性, 而这 3 中不同的版本恰恰突出了这个问题。随着业界担心的增加, 这个语言的标准化显然已经势在必行。

全文:

<http://developer.51cto.com/art/201202/315529.htm>

下面是本期编程语言排行榜的其他排名数据和趋势走向。

前 10 名编程语言走势图



下面是第 50 到 100 的编程语言排名

(Visual) FoxPro, ABC, Algol, Alice, Apex, bc, BlitzMax, Boo, CL (OS/400), Clean, Clojure, Dart, Dylan, Euphoria, Factor, Go, Icon, IDL, Informix-4GL, J, JavaFX Script, JScript.NET, Korn shell, LabVIEW, LabWindows/CVI, Ladder Logic, Lingo, LPC, Mathematica, Modula-2, MUMPS, NATURAL, Oberon, Occam, OpenCL, OpenEdge ABL, Oz, PowerShell, Q, REALbasic, REXX, S, S-PLUS, Scala, SuperCollider, VBScript, Verilog, VHDL, X10, xBase



**统**计表明如果一家电子商务网站每天可赚 10 万元,那么网站延迟一秒可能会带来每年 250 万元的收入。1 秒钟的延误会将顾客满意度降低 16%, 44% 的在线购物者会将他们不好的购物经历告诉朋友。页面加载的快慢不止是电脑网站,对手机用户、移动互联网用户更为重要!

2012 年 3 月编程语言排行榜 JavaScript 连升 2 名重新回到了第 8 名的位置,超越了 Perl 和 Python。可看出随着 HTML 5 的流行,JavaScript 现在极其重要,一些网站使用 JavaScript 来增添魅力,那么如何更快的加载你的 JavaScript 页面呢?Web 开发人员经常借助基于 JavaScript 的库或自定义方法,以便在自己的应用程序中能够实现拖放操作。幸运的是,HTML 5 本身内置了支持拖放的功能。那么如何在 ASP.NET 网站中使用 HTML5 的拖放功能呢?自从“Ajax”这个字眼在 2005 年崭露头角以来,它就改变了互联网的面貌,那么刷新页面,如何用 Ajax 构建更好的单页面体验呢?

若你关注这些,那么本专题不可错过!本系列专题文章结合实例帮助开发者了解和优化网站的性能。

51CTO 开发频道语

# 在ASP.NET网站使用HTML 5拖放

作者 / 布加迪编译

拖放操作在桌面应用程序中司空见惯。现在的 Web 应用程序也试图利用拖放操作的简易性和强大功能,提供改善的用户体验。

Web 开发人员经常借助基于 JavaScript 的库或自定义方法,以便在自己的应用程序中能够实现拖放操作。

幸运的是,HTML5 本身内置了支持拖放的功能。你使用拖放功能,可以拖动某个 HTML 元素,将它拖放到另一个 HTML 元素中。

在此过程中,还可以将数据从源元素传送到目标元素。如果把拖放操作与服务器端处理集成起来,你就可以提供丰富的用户体验。本文介绍了如何在 ASP.NET 网站中使用 HTML5 的拖放功能。

## 为 HTML 元素启用拖动功能

要使用 HTML5 的拖放功能,第一步是让一个或多个元素可以拖动。

为此,你只需要将 HTML 元素的可拖动属性设成 true。比如说,下面这行标记代码将 <DIV> 元素设成了可拖动元素:

```
<div class="myclass" draggable="true">Some content</div>
```

## 拖放事件

将一个或多个 DOM 元素标为可拖动元素只完成了一部分工作。想让你的拖放操作实际可以使用,而且对最终用户有视觉吸引力,就要处理某些事件。下面列出了这些事件:

事件	描述
dragstart	拖动操作开始时,该事件被触发。
drag	元素拖动时,该事件被触发。
dragenter	可拖动元素被拖动,并输入有效的拖放目标后,该事件被触发。
dragleave	被拖放到有效拖放目标的可拖动元素离开拖放目标后,该事件被触发。
dragover	可拖动元素被拖放到有效拖放目标上方后,该事件被触发。
drop	已拖动元素被拖放到有效拖放目标上面后,该事件被触发。
dragend	拖动操作结束后,该事件被触发。

你可以两种方法将事件处理函数连接到这些事件,即在 DOM 元素标记中,使用 onxxxx 语法,或者使用 JavaScript (或基于 JavaScript 的库,如 jQuery)。下列标记和代码显示了这两种方法。

```
<div class="myclass" draggable="true" ondragstart="OnDragStart" ondrop="OnDrop"></div>
$("div").each(function () {
    this.addEventListener('dragstart', OnDragStart, false);
    this.addEventListener('drop', OnDrop, false);
});
```

要注意上述代码如何使用 jQuery 代码中的 addEventListener() 方法来连接事件处理函数。

## 拖操作与放操作之间传送数据

大多数时候,拖动某个元素,然后把它拖放到另外某个元素上也需要在源元素与目标元素之间传送一些数据。为了完成这项数据传送任务,HTML5 提供了 DataTransfer 对象。下列表格列出了 DataTransfer 对象的一些重要属性和方法。

你通常会在 dragstart 和 drop 事件处理函数中使用 dataTransfer 对象的属性和方法。



## 如何在 ASP.NET 网站中使用 HTML 5 拖放功能 II

属性/方法	描述
<b>effectAllowed</b>	表明所允许操作的类型。可能的值是： none、copy、copyLink、 copyMove、link、linkMove、move、 all 和 uninitialized。
<b>dropEffect</b>	表明目前选择的操作的类型。如果操作类型得不到 effectAllowed 属性的支持，那么操作就失效。可能的值是：none、copy、link 和 move。
<b>setDragImage()</b>	设置拖动操作期间显示的特定元素。
<b>setData()</b>	设置所传送的特定数据。
<b>getData()</b>	检索之前设置的数据，以便进一步处理。
<b>clearData()</b>	清除之前存储的数据。

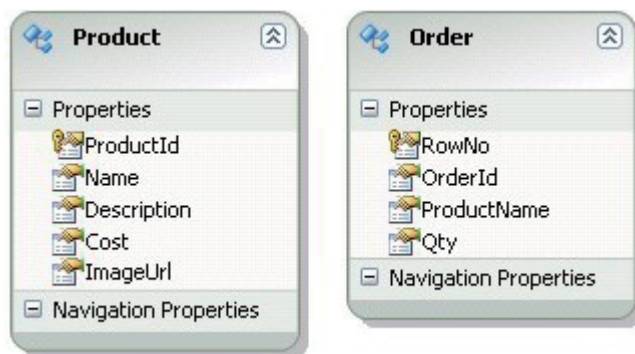
现在不妨把你到目前为止获得的信息放入到一个简单而实用的应用程序中。先建立一个新的 ASP.NET 网站。你将创建一个酷似下图的简单的 Web 表单：



简单的 Web 表单

注意：该实例在最新版的 Firefox 上经过了测试，但是应该也可以在最新版的其他主要浏览器上运行。正如你所见，Web 表单表示一辆简单的购物手推车。各产品由放在 DataList 控件里面的 DIV 元素来表示。这些产品可以拖放到

购物袋上。一旦所有需要的产品添加完毕，你可以点击“Place Order”按钮，即可将产品数据发送到服务器、下订单。上面这个例子使用了如下所示的 Entity Framework 数据模型。你可以从本文所附的代码下载链接( [http://developer.com/imagesvr\\_ce/6920/drag\\_drop\\_Code.zip](http://developer.com/imagesvr_ce/6920/drag_drop_Code.zip) ) 获得 SQL Server Express 数据库和数据模型。



该例子使用了 Entity Framework 数据模型。

上面所示的数据模型只包括基本的细节。在实际环境下的购物手推车系统中，你可以捕捉到多得多的细节。产品目录是一个 DataList 控件，它的 ItemTemplate 包括一个可拖动 <DIV> 元素。该 DIV 包装了某个产品的所有产品细节。

```
<asp:DataList ID="DataList1" runat="server"
DataSourceID="EntityDataSource1" RepeatDirection="Horizontal">
    <ItemTemplate> <div class="product"
draggable="true">
        <header><%# Eval("Name") %></header>
        <div><asp:Image runat="server" ID="img1"
ImageUrl='<%# Eval("ImageUrl") %>' /></div>
        <div><%# Eval("Description") %></div> <br />
        <div><%# Eval("Cost", "Cost : ${0}") %></div></div></ItemTemplate> </asp:DataList>
    <div class="bag"> <asp:Image runat="server"
ID="img1" ImageUrl="~/images/cart.jpg" /><br />
    <input id="Button1" type="button"
value="Place Order" /> <br />
    <input id="Button2" type="button"
value="Clear Cart" /> </div>
```

未完 查看网络原文：

[http://developer.51cto.com/art/201203/322148\\_1.htm](http://developer.51cto.com/art/201203/322148_1.htm)

# 刷新网站:用Ajax构建更好单页面体验

自从“Ajax”这个字眼在 2005 年崭露头角以来,它就改变了互联网的面貌。应用程序变得交互性更强了;桌面模式向 Web 迁移,而且在这个过程中不断嬗变;Web 作为一种平台已开始流行开来。



这个改变并没有影响网站的基本运作方式。当然,代码模块以异步方式装入;一些网站使用 Ajax 来装入通知;你在滚动浏览页面时,Facebook Like 按钮神奇地出现。典型的博客网站或宣传手册式网站(brochure site)提供了基于页面的体验;既然 Web 从本质上来说是一种基于页面的介质,为什么将事情搞得过于复杂呢?

但是移动应用程序的普及促使基于页面的交互迈上了一个新的台阶。导航区域待在原地不动时,屏幕可以顺畅地滑入滑出,不但提供令人愉悦的体验,还保留了上下文,进一步证实用户在信息层次体系中的位置。用户看到的是装入的动画,而不是屏幕与屏幕之间的一片空白。

## 不刷新网站

也许现在我们是时候为桌面 Web 采用这种方法了:不刷新网站(no-fresh website)。哪怕装入速度很快,整页刷新还是会影响上下文。借助 Ajax,我们可以在装入新内容的同时,让导航元素待在原地。我们可以通过进一步证实我们如何在网站的信息层次体系中穿梭的动画,装入新内容。而即使有了动画,这类操作也可能让人觉得比较轻便,因而鼓励访客继续导航——尤其是由于 Back(返回)操作触手可及。

诚然,这样一种网站比传统网站需要编写多得多的代码。但是这些代码有许多已经编好了:

jQTouch 和 jQuery Mobile 已经可以拿来多个 HTML 页面,创建动画的、基于 Ajax 的体验——总的来说,没有额外的 JavaScript 代码。它们为对桌面友好的框架提供了完美的基础。

我们不妨称之为 page.js。请恕我保留 pagejs.com(<https://github.com/schvenk/page.js>),将它指向全新的 GitHub 存储库,那样我们可以马上开始对它进行改动了。下面是其工作机理:

你可以按今天的方式来编写网站的 HTML,为针对特定标记的页面元素另外添加一点信息。(jQuery Mobile 使用 HTML5 数据属性,如 `<div data-role="content">`,以定义页面上的不同区域。)

网站首次装入时,page.js 读取和修改你的 HTML,使用可以移动的滚动元素,将随时准备滑出屏幕的那部分页面内容包装起来,并且相应调整你的内部链接。(如果性能至关重要,这可能出现在服务器端,缓存起来。)

访客点击内部链接时,装入动画在屏幕上显示的同时,新页面通过 Ajax 装入。当前页面滑出屏幕,新页面滑入屏幕。视情况而定,导航区域待在原地不动。

浏览器的 Back/Forward(返回/前进)历史记录使用 URL 的哈希部分(#后面的一切内容)保留下来。



## 刷新网站：用 Ajax 构建更好的单页面体验 II

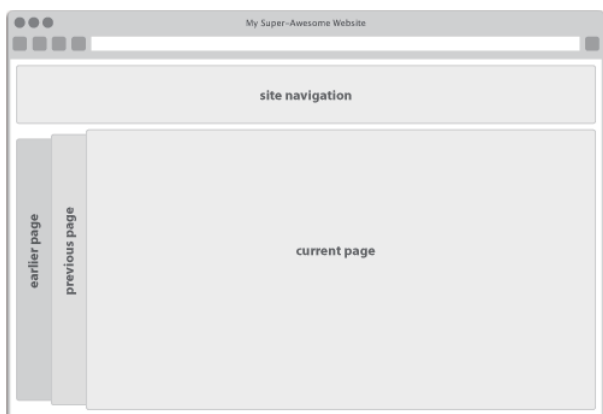
点击 Back 按钮, 之前的页面又滑入屏幕。你可以使用 Ben Alman 编写的 hashchange 插件( <http://benalman.com/projects/jquery-hashchange-plugin/> ), 在我的网站( <http://interfacethis.com/portfolio/> )上看看这个非常简单的例子。

### 新机遇

但这只不过是开了个头。单页面网站大有潜力。

当访客滚动浏览你的主页时, 何不在后台装入可能向下钻取的页面呢? 结合你的 Google Analytics 数据, 就可以确认当前页面之后最有可能访问的目的地。如果你喜欢, 也可以按访客的人口统计资料来加以细分。

新的导航模式因而成为了可能。之前的页面不需要一路滑出屏幕, 而是可以浏览旁边的部分, 以保留上下文, 从而创建一种现代的面包屑 (breadcrumb) 导航方式:



另外, 我们可以使用 HTML5 的离线功能, 为无法全部缓存的页面改进性能: 将导航元素及其他静态元素保留在客户端上, 然后使用那些缓存的元素来动态组装页面。

### 挑战

眼下, 对大多数来说门槛过高, 做不到这一点。唯有掌握丰富的专门知识, 并且投入大量的努力, page.js 才会成气候。

永久链接 (Permalink) 带来了挑战。基于 Ajax 的导航依赖 ULR 的哈希部分( <http://my-site.com/my-page#hash-component> )。它包含在书签里面, 但是并不传输到服务器端。所以, 哈希部分是永久链接的一部分, 但是它离不开能够支持 JavaScript 的客户端的合作, 又没有被谷歌及其他搜索引擎编入索引。但愿有一个解决办法: 谷歌已开发了一项标准, 好让 Ajax 链接能搜索到; 一旦使用了这项标准, 你的不刷新网站就能搜索到。而解决办法变得更好了: 许多浏览器在开始实施 HTML5 History API (应用编程接口); 相比之下, 这套 API 解决这个问题要漂亮得多。所以你现在只要用一点 JavaScript 代码, 就可以实现, 将来可以大大减少 JavaScript 代码。

从框架层面来看, 这些挑战可以应对的, 也是可以克服的。所以, 第一步是创建 page.js, 然后将它交到网站制作者的手里。之后就等着坐收其成了。方块

网络原文:

<http://developer.51cto.com/art/201203/321305.htm>

### 开心一笑

考试时和老师在写字时最喜欢问的问题!



# 性能优化：更快速加载你JavaScript

作者 / 布加迪编译

JavaScript 现在极其重要。一些网站使用 JavaScript 来增添魅力；如今的许多 Web 应用程序依赖它；其中一些甚至是完全用 JavaScript 编写而成的。我在本文中 will 指出一些重要的规则，介绍如何使用你的 JavaScript、使用哪些工具以及你将从中得到什么好处。

## 确保代码尽量简洁

不要什么都依赖 JavaScript。不要编写重复性的脚本。要把 JavaScript 当作糖果工具，只是起到美化作用。别给你的网站添加大量的 JavaScript 代码。只有必要的时候用一下。只有确实能改善用户体验的时候用一下。

## 尽量减少 DOM 访问

使用 JavaScript 访问 DOM 元素很容易，代码更容易阅读，但是速度很慢。下面介绍几个要点：限制使用 JavaScript 来修饰网页布局，把针对访问元素的引用缓存起来。有时，当你的网站依赖大量的 DOM 改动时，就应该考虑限制你的标记。这是改用 HTML5、舍弃那些原来的 XHTML 和 HTML4 的一个充分理由。你可以查看 DOM 元素的数量，只要在 Firebug 插件的控制台中输入：  
`document.getElementsByTagName('*').length`。

## 压缩代码

要提供经过压缩的 JavaScript 页面，最有效的办法就是先用 JavaScript 压缩工具对你的代码压缩一下，这种压缩工具可以压缩变量和参数名称，然后提供因而获得的代码，使用了 gzip 压缩。

是的，我没有压缩我的 main.js，但你要检查

有没有未经压缩的任何 jQuery 插件，别忘了压缩。下面我列出了压缩方面的几个方案。

◆ YUI 压缩工具(我的最爱，jQuery 开发团队就使用它)，初学者指南(<http://www.slideshare.net/nzakas/extreme-JavaScript-compression-with-yui-compressor>)、第二指南(<http://vilimpoc.org/research/js-speedup/>)和官方网站(<http://developer.yahoo.com/yui/compressor/>)。

◆ Dean Edwards Packer (<http://dean.edwards.name/packer/>)

◆ JSMIn (<http://crockford.com/JavaScript/jsmin>)

GZip 压缩：其背后的想法是，缩短在浏览器和服务器之间传送数据的时间。缩短时间后，你得到标题是 Accept-Encoding: gzip, deflate 的一个文件。不过这种压缩方法有一些缺点。它在服务器端和客户端都要占用处理器资源(以便压缩和解压缩)，还要占用磁盘空间。避免 eval()：虽然有时 eval() 会在时间方面带来一些效率，但使用它绝对是错误的做法。eval() 导致你的代码看起来更脏，而且会逃过大多数压缩工具的压缩。

## 加快 JavaScript 装入速度的工具：Lab.js

有许多出色的工具可以加快 JavaScript 装入的速度。值得一提的一款工具是 Lab.js。

借助 LAB.js (装入和阻止 JavaScript)，你就可以并行装入 JavaScript 文件，加快总的装入过程。此外，你还可以为需要装入的脚本设置某个顺序，那样就能确保依赖关系的完整性。此外，开发者声称其网站上的速度提升了 2 倍

## 性能优化: 如何更快速加载你的 JavaScript 页面 II

### 使用适当的 CDN

现在许多网页使用内容分发网络(CDN)。它可以改进你的缓存机制,因为每个人都可以使用它。它还能为你节省一些带宽。你很容易使用 ping 检测或使用 Firebug 调试那些服务器,以便搞清可以从哪些方面加快数据的速度。选择 CDN 时,要照顾到你网站那些访客的位置。记得尽可能使用公共存储库。

### 面向 jQuery 的几个 CDN 方案:

◆ <http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js>——谷歌 Ajax, 关于更多库的信息请参阅 <http://code.google.com/apis/libraries/devguide.html#Libraries>。

◆ <http://ajax.microsoft.com/ajax/jquery/jquery-1.4.2.min.js>——微软的 CDN

<http://code.jquery.com/jquery-1.4.2.min.js>——Edgecast (mt)。

### 网页末尾装入 JavaScript

如果你关注用户,用户因互联网连接速度慢而没有离开你的网页,这是一个非常好的做法。易用性和用户放在首位,JavaScript 放在末位。这也许很痛苦,但是你应该有所准备,有些用户会禁用 JavaScript。可以在头部分放置需要装入的一些 JavaScript,但是前提是它以异步方式装入。

### 异步装入跟踪代码

这一点非常重要。我们大多数人使用谷歌分析工具(Google Analytics)来获得统计数据。这很好。现在看一下你把你的跟踪代码放在哪里。是放在头部分? 还是说它使用 document.write? 然后,如果你没有使用谷歌分析工具异步跟踪代码,那也只能怪你自己。

这就是谷歌分析工具异步跟踪代码的样子。我们必须承认,它使用 DOM,而不是使用 document.write,这可能更适合你。它可以在网页装入之前检测到其中一些事件,这非常重要。现在想一想这种情况,你的网页甚至还没有装入,所有用户都关闭了网页。已找到了解决页面浏览量错失的办法。

没有使用谷歌分析工具? 这不是问题,今天的分析工具提供商大多允许你使用异步跟踪。

### Ajax 优化

Ajax 请求对你网站的性能有重大影响。下面我介绍关于 Ajax 优化的几个要点。

### 缓存你的 ajax

先看一下你的代码。你的 ajax 可以缓存吗? 是的,它依赖数据,但是你的 ajax 请求大多应该可以缓存。在 jQuery 中,你的请求在默认情况下已被缓存,不包括 script 和 jsonp 数据类型。

### 针对 Ajax 请求使用 GET

POST 类型请求要发送两个 TCP 数据包(先发送标题,然后发送数据)。GET 类型请求只需要发送一个数据包(这可能取决于你的 cookie 数量)。所以,当你的 URL 长度不到 2K,你又想请求一些数据时,不妨使用 GET。

### 使用 ySlow

说到性能, ySlow 既简单,又极其有效。它可以对你的网站进行评分,显示哪些方面需要改正,以及应关注哪些方面。■

**另外支一招: 把你的 JavaScript 打包成 PNG 文件** (未完,查看网络原文:

<http://developer.51cto.com/art/201203/323057.htm>



## ■ 编者按

以下,开始本系列的第6篇——设计模式。而实际开发中常用的设计模式更是屈指可数,Singleton, Factory, Facade, Active Record、Provider 等等。

# 宅男程序员给老婆计算机课程之5:设计模式

设计模式,应该是很多ED心目中牛B的编程方式。

上回说到ED的好书POEE,实际上便是一本专门讲企业开发中使用的设计模式中的书。

设计模式,并不多,基本上看完GoF的这边《Design Pattern》便可以有足够了解了。

而实际开发中常用的设计模式更是屈指可数,Singleton, Factory, Facade, Active Record、Provider 等等。

就那么几个,花花功夫,仔细了解一下几个,然后在实际编码中应用一下,便可算是掌握了。

设计模式,并不难。

它是开发中非常必要的知识,实际上,是非常基础的知识,并不牛B。

开发的时候,需要时刻明确自己的目标:解决问题。解决问题才是最重要的。

设计模式的存在,是为了更好的维护、管理代码,或者是为了扩展性;绝对不可以为了设计模式而模式。

在Java程序中,为了模式而模式的现象蛮普遍的。

我猜想这是因为Java是一门工业语言,有大量的ED使用的缘故。

ED把设计模式的使用,当成是一种可以炫耀的编程技巧,或者说,他们遵从的编码规范中,要

求他们去使用某某设计模式。

至于为什么要使用设计模式,最常见的理由便是:为了将来可以XX,或者YY。

程序开发中,有一句名言:“Pre-mature optimization is the root of all evil”。

过早优化,是万恶之源。

非常多的情况下,将来预想中的XX或者YY;并不会发生。大部分代码,写了之后就会被丢弃掉,再也不会有人去维护。

当XX或者YY发生的时候,往往,都是会推倒重来。

除非你很牛B,只有牛到一定程度,才有可能对将来可能发生的情况做好合理的预测,并预留出改善、调整的空间。

但非常讽刺的是,为将来做设计的最好方法就是:什么都不做。

只有空白,才能够留下最大的发挥空间。

现在为将来可能发生的情况,做了编码,任何一行编码,都是很可能是在为将来添加限制、制造麻烦。

现在写下去的代码,将来,都是要被删掉的;能够不写,就不写。在任何时候,都应该保持代码简洁。 ■

本文未完,请看原文:

<http://developer.51cto.com/art/201202/318911.htm>

## ■ 编者按

表面上看来这只是一些基础的问题,当真正的了解了以后你就会发现:啊,原来是这么一回事!下文是几道 Java 谜题,不仔细分析就会犯错哦。

# 宅男程序员给老婆的课程之模版引擎

设计模式再“高级”一点,便是所谓的“框架”了。

从事 Web 开发,一般都会接触到 MVC 框架这个概念。

M: 也就是 Model,直接跟网站数据库相关。

V: 也就是 View,是网页的模版,跟显示数据相关。

C: 则是 Controller,相当于网站的业务逻辑。

MVC 也不仅仅是应用于网站开发,它的概念实际上植根于桌面软件,并且在手机软件开发上也有应用。

MVC 本身是一个设计模式,一个被验证过的,可用来很好归纳、管理代码的软件开发方式。

基于这样的设计模式,提供了很多相关的类库实现,则“设计模式”升级为“框架”。

MVC 的任何一个方面,扩展出去讲,都可以讲上几天几夜。

今天只讲 V。

传统的 ASP / PHP 网站开发,V 是很混乱的。

默认只有一种文件,html 与业务逻辑代码混杂在同一个文件;相当难以维护。

ASP.NET 相对于 asp 做出了很大改进,提出了 code-behind 的概念:默认将 html 的模版代码,以及 c# 或者 vb.net 的逻辑代码切分到两个不同的文件。

这样的方式算是有很大进步。

微软平台上做开发是比较苦的,微软掌控了整个开发平台的前进速度。

ASP 跟 PHP 在开始的时候,是相似的技术。有类似的便利,以及类似的麻烦。

微软推出了 .NET,推广了 code-behind 的模式;然后,所有的微软程序员都超着微软指定的这个方向去迈进。

ASP 被抛弃了,自从 ASP.NET 诞生之后,就不再有任何改进。

而 PHP,在开源世界中,则不断的得到各式各样的改进。

各种模版引擎层出不穷;不仅可以实现 code-behind 这样简单的模版、业务代码分割;很多还直接引入了 MVC 的概念;实现了三层分割。

而 ASP.NET,则长期止步于 web form 的 code-behind,在开源世界中的 MVC 方案大放光彩若干年后,才推出 ASP.NET MVC。

模版技术,最初的目的就是要把业务代码,也就是说,把获得数据的代码跟 html 分割。

在模版实现上,因此涌现了不少不同的设计哲学。

Python 的 Django 框架的模版,是一种典型。

它彻底的禁止程序员在模版中嵌入任何代码;模版中,只可以出现 html;以及一些跟业务逻辑无关的控制标签,如:

```
{% If XXXX %} foo {% else %} bar {% end %}
```

## 宅男程序员给老婆的课程之模版引擎 II

条件 XXXX,必须是一个数据值,不可以是一个复杂表达式、不可以包涵函数调用等等。

模版中,也不可以声明任何新的变量,下面的做法是被禁止的:

```
{% set i = 0 %} {% foreach item in items %}
{% i += 1%} <div>  {{ item }}  {% if i mod 2
== 0 %}  <hr />  {% end %}  </div> {% next
%} Django 的模版,从技术上彻底禁止程序员添加任何逻辑,强迫程序员必须在 controller 中去写各种逻辑,以确保模版内容的纯洁干净。
```

所以 Django 的模版,一般都非常简单,有很好的移植性,并且可以让网页设计人员直接编辑。

ASP.NET 则是另一种典型;虽然有了 code-behind,但是它没有对前端代码,以及后端代码做任何限制。

在前端 aspx 页面中,可以嵌入任意的逻辑代码,而 code-behind 的 code,为空白;这种伪“code-behind”的方式,跟原来的 asp 没啥区别。

ASP.NET 从框架本身,并不阻止程序员去做这样的事情,实际上,它还标榜它这样的特性:方便原有的 asp 项目直接升级到 .NET 的平台上。

也有另外一种奇葩的做法,前端 aspx 页面保持空白,然后在 code-behind 的 code 中去拼接所有的 html。这样的方式,ASP.NET 框架本身也不禁止。

只要 ASP.NET 程序员喜欢,没有什么不可以的。

ASP.NET 把对模版使用方式的选择权留给了程序员,如果程序员自律,他们可以按 Django 模版那样的方式去使用模版,并拥有 Django 一样的优点;如果程序员自律?!

在某些可以通过嵌入代码去快速处理的场景,ASP.NET 的模版也保留了程序员去 hack 的能力。

还有一些模版技术,则是折衷的(如 tornado 的模版):允许嵌入单行代码,如声明变量,调用函数等等;但是不允许整块、整块的业务代码出现模版中。

上述三种模版设计哲学,各有它们的道理,以及应用场景。

需要根据具体的业务、应用场景,才能说其中哪种比较合适。

开发人员的能力也是直接相关的,如果团队中,普遍不自律;缺乏将业务、模版代码分割、以提高代码可维护性的意识,那么 Django 的做法是最好的,它直接禁止去滥用模版,强迫他们去使用更好的开发风格;即便在某些场景下会更麻烦。

武断的认为任何一种模版设计哲学是“最佳”的想法是极其肤浅的。

各种成熟的模版技术,一般也都会有包括以下特性:

### 1. 嵌入

也就是说,编写各种可以复用的小模版块,然后供多个不同地方调用;比方说,用户头像(甚至名片)的显示。

具体页面不需要重复编写这些重复的模块。

并且,这些模块需要调整时,只需要修改一个地方,便可以在所有地方生效。

### 2. 继承

能编写一些基础模版,定义常见页面结构。

具体页面继承这些基础模版,便不需要重复编写那些结构代码。



## 宅男程序员给老婆的课程之模版引擎 III

同样的,当页面结构需要调整时,也是修改一处,所有生效。

### 3. i18n

网页模版的国际化支持是一个模版引擎是否成熟的表现。

如果没有,当网站需要同时提供多种不同语言支持的时候,会很麻烦。

成熟的模版,都会提供内置的支持。

因为网页模版实现实在是太多了,大家功能也都差不多,那么性能,也就成为了相当重要的比较指标。

有的模版,能够“编译“,渲染起来快些。

一般可以简单认为,功能越多的模版,性能会约低。有的模版,甚至将 i18n 的支持变成可配置的,不需要的时候就可以关闭,以提高性能。

也有的模版认为,写 `{% %} <%%> {{}}` 这样的符号太麻烦了,可以直接忽略,它可以自动聪明的识别 html,以及模版控制代码。简单的说,就是以极其华丽的方式,去方面程序员少打几个字符。

还有的模版,在实现嵌入功能的时候,还可以选择所依赖的 css / js 文件。

比方说,要显示用户的名片,需要引入 namecard.css; 那么,可以在 namecard 的模块文件中指定这个依赖,然后模版渲染的时候,自动把这个 css 的引用,放在 html 的头部。

直接在模块文件中写 namecard.css 的引用是很傻的,因为模块可以在模版中引用多次。重复引用同一个 css 文件是没有道理的。

种种模版功能细节,实际上,都是可以在没有模版支持的框架中去实现。

想想 PHP,它本来是非常简单的,默认只能能够在同一个文件中混杂逻辑与代码。

但一旦程序员有了追求,它也可以有模版实现。

模版不支持 i18n,程序员一般也是有办法在现有模版实现中添加相应的支持的。

并不复杂,关键是看程序员的态度;看程序员是否有把事情做得更好、更优雅的态度。

一般情况下,程序员选择去实现更多的模版功能的时候,必须先看看别人是怎么做的。

比方说,如果完全不知道什么是 gettext 就去自行实现模版的 i18n 功能,是非常 2B 的。

绝大多数情况下,程序员面临的问题,都不是自己独有的,一定是别人已经解决过的问题。

是否有足够的见识,有足够的知识广度,了解别人的解决同样问题的做法是程序员能力的表现。

是否有快速的搜索出类似的解决方案,也是能力的表现。

1. PHP 的 Smarty 模版的设计哲学是什么?
2. Perl 的 Mason 模版的设计哲学是什么?
3. 什么是 gettext?
4. 前端 Javascript 实现的模版中,目前最成熟的是哪个引擎?

更多文章,请点击:

<http://developer.51cto.com/art/201203/321936.htm>

# 软件开发基本原则之最佳项目

本章的范畴只限定在确定软件开发的基本原则,解析他们是如何影响开发计划的,同时提供参考信息。

“回顾一下被选为‘最佳项目’的十个软件项目,如果说有所发现的话,那就是——最佳的项目一定是建立在最佳的软件开发基础之上的。我们都知道软件开发基础对于优秀软件的作用,但差别在于大多数软件的基础薄弱,这样不可避免地使自己陷入麻烦之中”(Bill Hetzel 1993)

本章的范畴只限定在确定软件开发的基本原则,解析他们是如何影响开发计划的,同时提供参考信息。

## 管理的基本原则

管理原则由以下几部分组成:

判定产品规模(包括功能、复杂度和其它产品特性)

根据产品规模分配资源

制定资源计划

监控、引导资源以保持项目方向不会偏离

## 1. 项目估算和进度安排

一个运行良好的项目一般通过三个基本步骤来定制软件开发进度表。

首先估算项目规模大小

然后估算完成这样规模的项目需要付出的代价

最后基于这种估算定制项目进度计划

如果估算不准确就会降低开发效率,所以说

估算和项目进度计划是软件开发的基础。精确的估算时进行有效规划的必要前提,而有效的规划又是有效开发的必要条件。

## 2. 计划编制

计划一个软件项目应该包括以下活动:

项目估算和时间进度

确定项目需要多少人参与、需要什么样的技能、合适加入以及具体人选

确定项目组的运作方式

确定项目采用的生命周期模型

管理风险

确定项目策略(例如:如何控制产品的特色,是否需要购买部分产品组建)

## 3. 跟踪

跟踪是一个基本的软件管理行为。如果不跟踪一个项目就不能管理它,就不会知道计划是否被贯彻执行了,也不会知道下一步该做什么,同时也无法监控项目风险。有效的跟踪能使项目组在还有时间做点什么来改正错误的时候,尽早发现进度表上的问题。

制定了一个项目计划就要跟踪检查它是否在按计划进行,包括对它的进度、费用和质量等目标的检查。典型的管理级跟踪控制包括:任务列表、进展状况会议、进展报告、里程碑审查、预算报

## 软件开发基本原则之最佳项目 II

告以及走查管理等。典型的技术级跟踪包括：技术审查、技术审计和标志着里程碑是否完结的质量关口等。

### 4. 量度

老板问你：“我们能够在 9 各月内开发出这个产品吗？”——你怎么回答？！

为了使开发更有效，你需要具备软件量度方面的基本知识。你需要了解收集数据的尺度基准，包括应该要收集什么数据，如何获得这些数据。你还需要具备用来分析状态，质量和生产率的详细基准方面的知识。任何公司想要进行快速的开发就要收集这些基本的尺度，这样才可以知道他们的开发速度是否正在改善或后退。

### 技术基本原则

1984 年有关“现代程序设计实践方法——技术的基本原则”的一份研究，详细论述了不使用这些基本原则就不可能具有高的生产率的内容。

（不广泛地使用“现代程序设计实践方法”就无法具有高的生产率）

很显然，不采用现代程序设计实践方法的项目不可能具有高的生产率。但技术基本原则的应用，就其本身而言，不足以创造高的生产率。一些项目使用了大量现代程序设计实践方法，但是仍旧和那些完全没有使用该方法的项目具有一样的生产率。因此，注意技术的基本原则是很必要的，但却不足以达到快速开发的目的（例如犯了某些典型错误）。

### 1. 需求管理

研究数据：

典型项目平均会经历 25% 的需求变化，从而

至少产生 25% 的额外费用和时间。

一项针对 8000 多个项目的调查显示，导致项目推迟发布、超出预算、功能比预期减少的最重要的三个原因——缺乏用户的介入、不完善的需求分析和用户不断改变需求，都和需求管理有关。（Standish Group 1994）

一项软件工程研究所的调查也有相同的结论：超过半数的项目都遭遇过不充分的需求管理的麻烦。（Kitson and Masters 1993）

需求管理就是收集需求，把需求记录成文档、电子邮件、用户界面串连脚本、可实现的原型等形式，然后依此来跟踪设计和编码，并随时管理、修改需求，以适应项目后续的过程。

成功的需求管理取决于了解足够的不同的实践经验，以便能够为特定项目选择可借鉴一种。

需求管理的基础：

需求分析方法：包括结构分析、数据结构分析和面向对象分析

系统建模实践：如类图表、数据流图表、实体关系图表、数据字典符号和状态跃变图表

沟通实践：如联合应用开发、用户界面原型和常规会谈实践等

需求管理和其它生命周期类型的关系：如渐进原型、阶段交付、螺旋模型、瀑布模型和编码修正需求管理在两个方面对开发速度发挥着巨大的调节作用：

首先，正规的需求管理中，需求收集往往比其他软件开发活动完成得要从容些。如果能加快需求步伐而不伤害质量，就可以缩短总的开发时间。



## 软件开发基本原则之最佳项目 III

第二,正确地把需求摆在首位,往往要比被动地这样做所花时间少得多。一些需求管理实践基本原则能够减少需求变化的数量,其他开发实践的基本原则能够减少因需求改变而产生费用。

想象一下,如果把需求变化从 25% 减少到 10%,同时把每个需求变化导致的费用减少 5%–10%,那么综合的效果会怎样呢?

### 2. 设计

研究数据:

一个设计上的错误如果到系统测试时才发现,那么花费的修补时间要比它在设计阶段时被发现所花费的时间多 10 倍。(Dunn 1984)

设计是系统构建、项目进度计划、项目跟踪和项目控制的基础。

#### 体系结构和设计的基本原则:

主要设计风格:如面向对象设计、结构化设计和数据结构设计

基础设计概念:如信息隐藏、模块化、抽象、封装、聚合、耦合、层次、继承、多态、基本算法和基本数据结构

对典型挑战性事件的标准设计:包括异常处理、国际化、本地化、便携性、字符串存储、输入输出、内存管理、数据存储、浮点算法、数据库设计、性能和复用

对特殊领域应用程序设计的独有思考:例如财务应用、科学应用、嵌入式系统、实时系统、安全性要求高的软件等

架构安排:如子系统组织、分层结构、子系统通信方式和典型的系统架构

设计工具的使用

### 3. 构建

当构建开始时,项目成功与否大多就已经注定了。需求管理和设计对开发进度计划的调节作用比构建的调节作用大得多,这意味着小的波动可以导致进度的重大变化。

尽管构建是一个低层次的活动,但是它确实可以提供许多机会进一步改进时间效率低的任务或优化一些任务。例如,花时间对那些无需镀金的功能进行镀金;调试那些无用的多余代码,或者对那些并不知道是否需要优化片段尽心优化。

构建的基本原则:

编码实践:如变量和函数命名、版面布局 and 文档数据相关概念:如作用范围、持续和捆绑时间,特定数据类型的使用方针:如通用基础数据类型、枚举、常量、数组和指针,控制相关的概念:如组织整齐的代码、条件的使用、循环的控制、复杂度的控制、特殊控制结构的使用(`goto`、`return`、`递归`)断言和其它以代码为核心的错误检测方法。对例程、模块、类和文件代码打包的规则

单元测试和调试实践

集成策略:如增量式集成、大爆炸式集成和渐进开发

#### 代码优化策略和实践

与所使用的特定编程语言相关的其他事情

使用构建工具:如编译环境、群组工作支持、源代码控制、代码库和代码生成器。软件配置管理 (SCM) 是管理项目成果的一种实践方法,能使项目在全程中保持一致的状态。■

本文未完,更多内容参见:

<http://developer.51cto.com/art/201203/321813.htm>

# 微软发布SQL Server 2012 RTM版

作者 / 彭凡

北京时间 2012 年 3 月 8 日上午 8 时,微软发布 SQL Server 2012 RTM 版本。标志着 SQL Server 2012 已经可以进入生产领域了。

## 瞄准大数据处理领域



SQL Server 2012 在大数据处理方面有了更大的发展,为了进一步把 SQL Server, Microsoft Excel、PowerPivot for Excel 2010 与 Power View 等丰富的商业智能工具和其他非结构化数据连接起来。

微软宣布计划将在 2012 年上半年发布面向 Windows Azure 平台的一项基于 Apache Hadoop 服务的限量预览版。

更加可靠的数据处理,私有云也是 SQL Server 2012 的亮点

有不少专业数据处理人员对于 Excel 这样的产品嗤之以鼻,特别是在统计分析领域。

在统计计算精度上确实会有些许的偏差,所以微软会把上面提到的多项产品整合在一起。

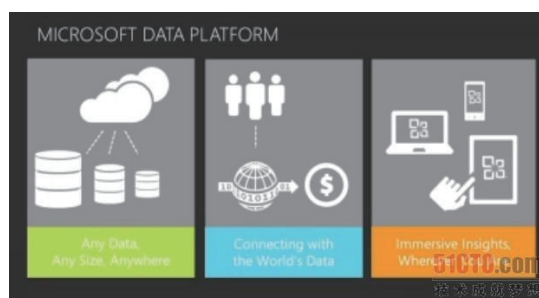
比如 Excel 虽然稍逊于 SAS 这样的专业软件,但与 SQL Server 的无缝连接可以让员工节约更多的时间,也有助于整个 IT 系统以一个更加紧密的整体工作。



## SQL Server 2012 的大步伐

更高、更快、更强,这已经被微软用来形容 SQL Server 2012 的强大。在可靠数据的基础上实现更快的数据查询,建立属于团队自己的云,BI 智能等等方面都是 SQL Server 2012 的亮点。

## 任何数据、任何大小、任何地方



## 微软数据平台

这次微软发布 SQL Server 2012 时提到 SQL 2012 将做到 “Any Data, Any Size, Any Where”。如上图所示,SQL Server 2012 将连接全世界的数据,从金融到人际交互。SQL Server 将作用于用户的个人电脑和手持智能终端,如手机和平板电脑。这些数据都将透过 SQL Server 2012 连接起来。

Windows 8 的主要改进就在于无缝链接 PC 机与其他手持智能终端,而 SQL Server 2012 将从后台数据链将这种连接更加牢固。■

## SQL Server 2012 RTM 下载链接

<http://www.microsoft.com/sqlserver/en/us/default.aspx>

## ■ 编者按

Java 是一种杰出的产业开发语言,这是因为它带来了伟大的统一和对事实上以前并不存在的重要标准的关注。但是和所有语言一样,Java 将来也会褪色。依据我做的超越 Java 的研究,一个重复出现的主题是有越来越多的人相信 Java 已不再足够的有效率。以下一组技术可以使你更有效率。

# 有可能挑战 Java 优势的四种技术

Java 是一种杰出的产业开发语言,这是因为它带来了伟大的统一和对事实上以前并不存在的重要标准的关注。但是和所有语言一样,Java 将来也会褪色。依据我做的超越 Java 的研究,一个重复出现的主题是有越来越多的人相信 Java 已不再足够的有效率。以下一组技术可以使你更有效率。他们不是对所有的工程都适合,但当被应用于适合的工程时,他们是优秀的工具。

## 1、动态语言

动态语言可以比像 C++ 或 Java 这样的静态语言更加有效率。他们可以让你用更少的语言表达更多的意思。这里,我会关注现在新出现最流行的动态语言 Ruby。用 Ruby 的 "Hello, World" 和 Java 的作一个比较:

```
puts "Hello, world."
```

这显然既简单又明了。你不需要写一些其他的代码去做这件事。下面是用 Java 语言的描述:

```
class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!")  
    }  
}
```

在 Java 中,类型是静态的。这就代表了编译器要检查所有的类型;你必须建立和编译一个完整的程序。在 Ruby 中,类型是动态的,所以你不

需要去声明他们,你可以马上编写和运行他们。下面是用两种语言描述的 Fibonacci 序列:

```
First, Ruby:x1, x2 = 0, 1  
  
//110.times do puts x2 x1,  
  
x2 = x2, x1+x2  
  
//4end
```

注意到在第一、四行同时声明两个变量,让你可以简洁地表达两种不同的声明形式。另外,注意到 10 是一个对象,它支持方法,如: times。再另外,在 do 和 end 之间是一个代码块。Ruby 代码块可以让你把代码块传入方法。。这种技术导致了难以置信的效率和简洁的代码。接下来,看一看用 Java 实现的代码:

```
class Fib {  
    public static void main (String args[]) {  
        int x1 = 0;  
        int x2 = 1;  
        int total = 1;  
        for (int i=0; i<10; i++) {  
            System.out.println(total);  
            total = x1+x2;  
            x2 = total; } }  
}
```

本文未完,请阅读原文: ■

<http://developer.51cto.com/art/201202/319101.htm>

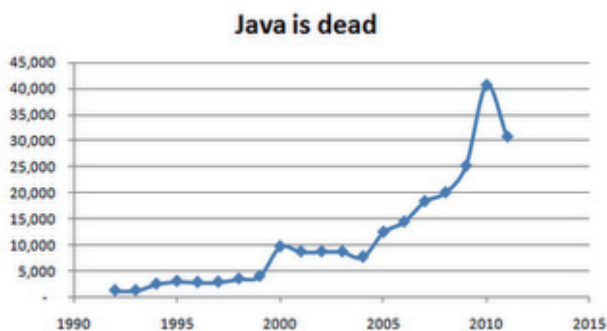


# “Java已死” 简史

“Java 已死” 已经成为一个著名的口水战,挑起这场议论的人希望能有一种更酷的语言出现、流行。它有可能是 D 语言,4GL,Ruby,Groovy。

“Java 已死” 已经成为一个著名的口水战,挑起这场议论的人希望能有一种更酷的语言出现、流行。它有可能是 D 语言,4GL,Ruby,Groovy。

有这种愿望的人看起来在不断的增加。

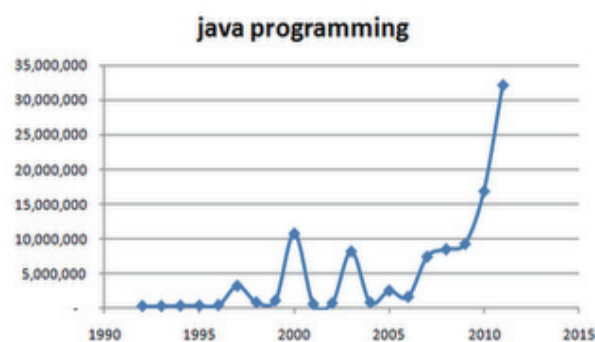


这显示的是由 Google 估算出来的关于“Java 已死 (Java is dead)” 的访问量按年的统计数

而跟“Java 已死” 内容扯上关系最多的语言是什么呢? 我在 Google 里查询“Java 已死” 和语言, 前十二种是

D	35000
C	30300
PHP	25800
C++	21300
Python	20000
javascript	19000
C#	18900
ruby	18100
perl	13700
Cobol	11800
Visual Basic	10300
groovy	10100

然而,把这个数据跟关于按年的“Java 已死” 的访问量相比较,这显示人们对 Java 的兴趣仍然在增加。



最近的一个调查也证实了这点,在它的列表里不仅显示 Java 是最流行的编程语言,而且向下排到第六位 (C#),你才能找到一种比较新的语言。看起来,Java 的主要竞争者应该是 C, C++, Python, 和 PHP( 同龄 )。■

51CTO 精品杂志推荐《Linux 运维趋势》

《Linux 运维趋势》是由 51CTO 系统频道策划、针对 Linux/Unix 系统运维人员的一份电子杂志,内容从基础的技巧心得、实际操作案例到中、高端的运维技术趋势与理念等均有覆盖。





■ 今天,我们宣布了 Java FX 1.2 和 Java FX 1.3 将要达到寿命末期,于 2012 年 12 月 20 日结束。更具体地说, Oracle 服务器将不再提供对这些版本的 Java FX 运行时的访问。

## JavaFX1.2和1.3寿命将要结束

早在 2010 年 9 月,在 JavaOne 会议期间,甲骨文公司就宣布终止 JavaFX 脚本(JavaFX Script)语言。从 Java FX 2.0 版本开始,甲骨文公司还承诺通过 Java API 公开所有的 JavaFX 功能。

今天,我们宣布了 Java FX 1.2 和 Java FX 1.3 将要达到寿命末期,于 2012 年 12 月 20 日结束。更具体地说, Oracle 服务器将不再提供对这些版本的 Java FX 运行时的访问。

由于 JavaFX 1.X 在产品线使用了特殊的部署架构,这意味着 JavaFX 1.2 或 JavaFX 1.3 应用程序的用户将无法下载 JavaFX 运行库。因此,我们强烈建议使用 JavaFX 1.x 的开发者和公司尽快将应用迁移至 JavaFX 2.0。目前可用在 Windows, Mac OS X 和 Linux。■

下载地址:

<http://www.oracle.com/technetwork/java/javafx/downloads/devpreview-1429449.html>



■ 根据原定计划, JDK 6 和 Java SE 6 将在今年 7 月份终结,现在该日期已经被推迟到了 11 月份。甲骨文称,之所以推迟,是为了给予开发者更多的时间来过渡到 JDK 7。

## Java SE 6生命将今年11月终结

甲骨文近日发布了一则公告,推迟了 Java SE 6 的终结期限。

根据原定计划, JDK 6 和 Java SE 6 将在今年 7 月份终结,现在该日期已经被推迟到了 11 月份。甲骨文称,之所以推迟,是为了给予开发者更多的时间来过渡到 JDK 7。

更多信息可以查看 Java SE 的支持路线图。

甲骨文称,一旦一个 Java 产品已经发布了三年,并且其继任者也已经发布了一年,那么该版本的寿命将会被终结。Java 7 是在 2011 年夏天发布的,而该日期距离 Java 6 的发布已经过去了 4 年半的时间,甲骨文又将在明年夏天发布 Java 8,看来 Java 6 已经到了生命的尽头了。

对于需要长期支持的开发者,甲骨文提供了一个商业版本。■

## Facebook版《愤怒的小鸟》为何选用Flash

你是不是想知道为 Facebook 开发《愤怒的小鸟》的首席 Flash 程序员究竟是何方神圣？不妨会一会来自《愤怒的小鸟》游戏开发商芬兰 Rovio 公司的 Ville Koskela。他在这篇独家采访中披露了 Rovio 的开发团队幕后采用的一些开发技巧。



问：Ville，你好！看到 Facebook 版《愤怒的小鸟》拥有种种新颖而出色的社交功能，真是让人惊叹。

Ville：确实如此！过去的几周忙得不可开交，因为对于像《愤怒的小鸟》这样广受欢迎的品牌，我们确实不得不确保第一个版本极其稳定。

问：这款游戏使用了 Flash Player 11 中新的 Stage3D 应用编程接口 (API)，之前你为这款游戏使用的是哪种 2D 框架？

Ville：现在使用 Starling 框架，它既提供了界面等 Flash 显示对象，又提供了相当好的性能。

问：你是 Starling 直接就拿来使用吗？还是说针对具体的使用场合，对某些方面进行了调整？

Ville：：我们一开始使用的初始版 Starling 的性能不如我们想象的那么好，但是由于它是采用简化的 BSD 许可证来发行的，所有源文件都随时可供使用，因而我们得以在诸多方面对它进行了优化。我一直在自己的博客中撰写这些优化方面的文章；现在大多数优化包含在最新版的 Starling 中。Daniel 也添加了另外几处调整。

问：软件渲染方面又怎么样？能不能透露几招，以便开发人员可充分发挥软件渲染的功能？

Ville：从我们的游戏收集而来的性能数据表明，就软件渲染而言，最大帧速率预计只有硬件渲

染的三分之一。这意味着，如果你的游戏要求很高的帧速率，那么使用软件渲染时，可能会丢失一些图形细部。就 Facebook 版本的《愤怒的小鸟》而言，我们现在丢弃了最后面的背景层，还调整了渲染质量，如果游戏使用软件渲染来运行的话。

问：你是否为每一个元素使用不同的图像，或者精灵图 (sprite sheet)？为什么？

Ville：：我们使用自己实施的精灵图，我们在同一张图上合并尽可能多的精灵：2048 × 2048 像素纹理大小方面的限制允许可以使用多少个精灵，我们就使用多少个精灵。这加快了 Stage3D 渲染速度，因为之后，共享同一基本纹理的 Starling 精灵就可以成批渲染。

问：你是否把 DisplayList 元素覆盖到 Stage3D 上，还是说整个游戏就用 Stage3D？如果这样，你覆盖什么？

Ville：只有游戏界本身才使用 Stage3D 图形。我们决定使用传统的 Flash 显示对象来开发用户界面的所有元素，因为使用我们的图形设计师们熟悉的 Flash 集成开发环境 (IDE)，就很容易为它们赋予动画效果。Flash 11 也能很好地处理显示对象覆盖，它们其实并不影响帧速率。

问：你如何处理设备丢失（借助 Stage3D 来处理）？这是否意味着，如果有人玩游戏时去吃午饭，PC 就会锁住屏幕？

## Facebook 版《愤怒的小鸟》为何选用 Flash II

Ville: 我们为这个事件实施了自己的处理程序,因为 Starling 本身在默认情况下并不支持。渲染设备丢失后,Context3D 的 driverInfo 字符串就会变成“Disposed”(释放),那样针对每次帧更新,我们就能进行这方面的检查。要是设备丢失了,我们可以立即从渲染返回。当 Flash 能够创建新的渲染上下文时,就会发送 Event.CONTEXT3D\_CREATE 事件。我们只要侦听这个事件,然后下一次渲染调用时,对所有当前的纹理及其顶点和索引缓冲器进行重新初始化。幸好,即将发布的 Starling 版本也可以处理设备丢失问题,那样开发人员再也不需要自行实施这种检查机制了。

**问: 你能不能对我们讲讲你与 Starling 结合使用的粒子引擎?**

Ville: 我们使用的是经过调优的 Starling 粒子引擎扩展。眼下,它用于“Suft and Turf”这一集中的海浪动画。粒子引擎本身结合了操作脚本代码和基于图形处理器 (GPU) 的 Stage3D 渲染。

**问: 你使用哪种物理引擎? 你使用默认的设置还是某些特别的设置?**

Ville: 与其他所有版本的《愤怒的小鸟》一样, Facebook 版《愤怒的小鸟》同样使用 Box2D, 只是某些方面作了细微调整。

**问: 你是否用某种代码混淆软件为最终的 SWF 文件确保安全?**

Ville: 是的,的确如此。

**问: 你在开发这款游戏时,是不是要考虑到 Facebook 的某些独特方面?**

Ville: 首先,就 Flash 11 和 Stage3D 而言,你必须使用窗口模式 (wmode) 为“direct”,这意味着在一些浏览器上,HTML 覆盖会在 Flash 内容下面可以得到渲染。为了处理这个问题, Facebook 提供了 javascript API,我们把 Flash 内容换成了在网页上屏幕截取。我们还想让初始游戏下载的速度尽可能快,以确保用户不至于还没有运行游戏、就关闭了游戏。为此,只有在需要时,我们才为关卡实时下载额外的游戏资产。



Ari(Rovio 高级 Flash 程序员) 和作者

**问: 你总体上觉得 Facebook 版《愤怒的小鸟》如何? 有没有计划为游戏增添新的功能特性?**

Ville: 我认为,这个项目取得了很大的成功,我们从 Adobe 和 Facebook 都得到了大力支持,这给我们的这个项目帮了大忙。在最终用户看来,连初始版都没有太多的软件缺陷 (bug); 就算那些软件缺陷现在大部分也得到了修复。说到将来,其他的我确实讲不出太多,只能每周至少一两次回过头来关注游戏,看看有没有新的功能。■

网络原文:

<http://developer.51cto.com/art/201203/323368.htm>

查看更多请访问:

<http://developer.51cto.com/>



# IT裁员为何总是测试工程师最受伤?

上周微博传言腾讯将会大幅度裁撤测试部门的员工,原来的测试工作,而改由外包人员担任,而且本次裁撤测试岗位的动作,不是只针对一个部门,可能整个公司所有产品线的测试岗位人员全部换成外包公司的劳务派遣人员,可能裁员达 300 人之多。



事后腾讯员工吴凯华微博回应,腾讯即将启动实习生招聘工作,测试岗位将有上百名需求,让散播腾讯测试裁员的谣言者闭嘴。传言卸下帷幕,但测试工程师的心却是一直悬在空中,有网友称,其老婆是做软件测试的,经历了 4 次工作变动,全部是因为公司裁员,而且无一例外都是先拿测试下手!软件测试工程师简单的说是软件开发过程中的质量检测者和保障者,那么 IT 公司裁员为何总是软件测试工程师最受伤呢?

## 1. 测试工程师是项目团队中的服务员

尽管没有任何 IT 背景的人也可以从事测试工作,但是一名要想获得更大发展空间或者持久竞争力的测试工程师,则计算机专业技能是必不可缺少的。但无论你是多么高级的测试工程师,无论测试需要的工具多么复杂,测试步骤多么冗长,都要明白测试工程师在软件项目开发中始终都是扮演服务员的角色,这是由测试工作的特点决定的。测试工程师服务的主要对象是软件的用户、项目经理、程序员。当一个酒席完成时,厨师长(项目经理)和大厨(程序员)可以休息等下一单,但除掉少数核心的服务员可以保留,其他完全可以散去,毕竟服务员入行要求较低,短时间可以急招,让核心服务员加一培训即可上岗满足最基础的需求,实在不行大厨也可以兼任。

## 2. 降低企业成本与风险

劳务派遣的测试人员薪酬普遍比正式员工要低,据业内人士称,以国内大型互联网公司为例,劳务派遣的测试人员可能比企业自己的测试人员少拿 3K-5K,每名测试人员一年即可节省 4-5 万人民币,一百名即可节省几百万,几百人就是几千万。另外一般大型的 IT 企业年终都有年终分红,而劳务派遣人员可能没有或者只有很少,再加上节省的社保和办公费用,一年或可节省上亿成本。

## 3. 专业问题让专业人士解决

早已经有很多的大型公司,例如谷歌、微软、华为、IBM 等国际软件旗舰企业,利用第三方的专业软件测试公司,在产品发布前对软件进行一系列的集成测试和系统测试,即保证了测试工作的全面性,又节省了人力、物力的开销。最重要的是,测试结果往往好于这些软件企业最初的预期,效果非常令人满意。

### 对于测试工程师的职业规划,网友建议:

◆ 测试工程师可提高自身技能,提高对用户体验等方面的判断水平,向更高层次发展。

◆ 测试工程师工作几年后可转产品经理,因为测试需要你对一个产品的整体架构、人机交互、系统性能等有充分的了解。这一切恰好也是产品经理们工作的重点。■

# 国内主流开放平台生态系统建设情况调研报告

开放平台生态系统中,开发者收入“二八原则”愈发显现,占大多数的开发者依然难以通过开放平台获得收入,影响第三方开发者收入的主要原因是什么呢? 开放平台提供的政策是否能够有效提高开发者的收入? 如何有效提高开发者的收入呢?

互联网企业中腾讯、百度、奇虎 360、新浪、淘宝、人人网、开心网等均已宣布开放平台,并开始积极布局。继 facebook 之后,中国互联网的开放大门陆续打开,开放平台大势已经不可阻挡,平台模式并非一种新兴商业模式,社会化大生产导致人类分工合作,通过分工和协作,提高生产效率,节约生产费用,让不同的企业提供他们擅长的服务,作为这个生态系统环节的平台运营商、开发者、用户、广告主和广告公司、第三方服务商都将参与其中,通过开放,获得最大的利益。

但是,开放平台生态系统中,开发者收入“二八原则”愈发显现,占大多数的开发者依然难以通过开放平台获得收入,影响第三方开发者收入的主要原因是什么呢? 开放平台提供的政策是否能够有效提高开发者的收入? 如何有效提高开发者的收入呢?

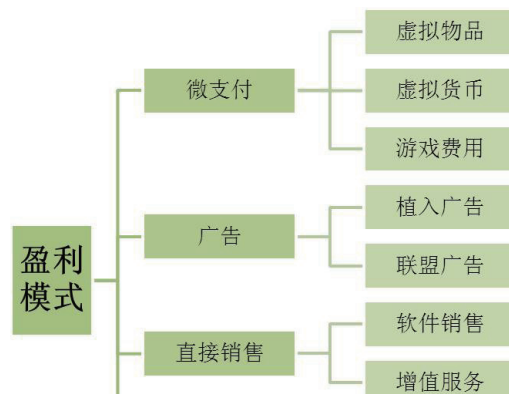
根据上面的问题,51CTO 调研中心根据公开资料,对上面的问题进行深入探讨,希望能够引起开发者思考,并在开发者探索盈利模式过程中提供有价值参考。

调研报告下载地址: <http://down.51cto.com/data/342325>

国内主流开放平台生态系统建设情况



## 开放平台的主要赢利模式



## 中小开发者沦为“二八原则”的牺牲品

根据最新调查数据,64.5% 的国内开发者处于亏损状态,部分开发者没有收入,其中暂无收入的为 22.4%

本文未完,请阅读原文: ■

<http://developer.51cto.com/art/201203/322174.htm>

# 三大移动Web框架哪个适合你?

本文由世界知名的 PHP 开发人员 W.Jason Gilmore, 在 developer 上撰写的一篇比较目前主流移动 Web 开发框架的文章。文章探讨了 jQuery Mobile、jQTouch 和 Sencha Touch 究竟哪个更适合开发者。

经常浏览 Developer.com 的读者也许注意到了这点: 我最近特别关注 JavaScript 驱动的移动应用开发解决方案, 先后撰文介绍了众多的相关话题, 包括 jQuery Mobile、jQTouch、Sencha Touch 和 PhoneGap。我得承认, 要搞清楚这一批前沿技术并非易事, 于是我这回概述一下这些主要的移动开发解决方案的主要特性和优点。但愿这篇概述多少可以帮你确定哪一种解决方案最适合你项目的具体要求。



## jQuery Mobile 适合你吗?

差不多二十年来, 处理跨浏览器的 HTML 和 CSS 不一致问题一直让许多开发人员颇为苦恼。而 Ajax 的出现更是加剧了这些问题, 因为开发人员现在要处理新的问题: 无数与 JavaScript 有关的浏览器不兼容问题和异常现象。后来出现了 jQuery: 由于可以对基本的 JavaScript 任务进行强大的抽象处理, jQuery 大大减轻了编写复杂的跨

浏览器代码方面的压力。难怪 jQuery 一下子火了起来。

眼下移动应用开发人员面临可能更艰巨的挑战, 这不仅是由于要处理各种各样的移动浏览器不兼容问题, 还由于需要设计合理布局, 以便适合移动设备所固有的各种外观尺寸和有限资源。这时候, jQuery Mobile 应运而生, 它自诩是“一种针对触摸而优化的 Web 框架, 面向智能手机和平板电脑”, 可以充分发挥 jQuery 和 jQuery UI 的优势。

虽然 jQuery Mobile 目前仍是测试版 (第二个测试版在去年 8 月 3 日宣布发布), 但已经支持数量多得惊人的移动平台, 包括安卓、黑莓、iOS 和 Windows Mobile 等其他平台。同样出色的是, 它还支持众多的窗口组件: 由于 jQuery Mobile 对渐进增强很重视, 开发人员可以利用这些窗口组件, 轻松创建统一的、跨移动设备的界面。务必将演示程序装入到众多不同的设备上 (包括你的桌面浏览器), 领略一下界面是如何变化以适合目标设备的。此外, 务必从桌面浏览器里面来查看演示程序的源代码; 连刚入门的 jQuery 用户也会注意到该程序的代码很注重全面隔离页面布局和 JavaScript 驱动的功能。你会看到, 整个应用程序都在一个文档里面, 每个“页面”及其组成部分使用与自定义属性有关的 DIV 来定义。

## 三大移动Web开发框架哪个适合你？II

比如说：

```
<div id="home" target="_blank">
<h1>WJGilmore, LLC</h1> </div> <div>
<p>Check back regularly for the latest
information about Jason's books!</p></div>
<div><h1>Copyright © 2011 W.J. Gilmore,
LLC</h1>
</div>
```



### jQuery Mobile 的主要特性

虽然使用 jQuery Mobile 的理由有多个,但是下面三个理由显得尤为突出:

jQuery 开发人员使用 jQuery Mobile 时会觉得驾轻就熟,那是由于两者使用共享的代码库。

如果目标用户群并不局限于使用某一种设备,关注跨平台兼容性这个优点很理想。

社区和企业的广泛参与促使像 Dreamweaver CS 5.5 这些广受欢迎的开发产品和像非常出色的 Photo Swipe 这一批值得关注的插件内置了对 jQuery Mobile 的支持。

### jQuery Touch 适合你吗？

那边厢, jQuery Mobile 力求提供在众多移动

设备上兼容的功能;这边厢, jQuery Touch 项目而是将大部分精力投入到了支持基于 WebKit 的浏览器 (Chrome 和 Safari) 的设备上。因而, jQuery Touch 应该与旨在部署到 iPhone 和安卓设备上的 Web 应用程序结合使用。支持针对 WebKit 而优化的动画以及旨在与移动设备的式样融为一体的多个主题。

要熟悉 jQuery Touch 的功能,最容易的办法就是通过你的移动设备或 WebKit 浏览器,仔细研究演示网站。我尤其建议查看演示程序的源代码;你会看到,与 jQuery Mobile 一样, jQuery Touch “页面”也使用一系列 DIV 元素和相关属性来组装,设计人员对这种方法应该会很熟悉。



### jQuery Touch 的主要特性

虽然使用 jQuery Touch 的理由有多个,但是下面两个理由显得尤为突出:

虽然 jQuery Touch 同样基于 jQuery,但是其侧重点主要是为基于 WebKit 的浏览器提供经过优化的支持。因而,如果你打算一心锁定使用基于 iOS 或安卓的设备用户,那么 jQuery Touch 可能比较合适。

与 jQuery Mobile 一样, jQuery Touch 也采用了以 HTML 和 CSS 为中心的方法来开发页面,



### 三大移动Web开发框架哪个适合你? III

使用 JavaScript 专门致力于添加事件、动画及其他功能。如果你在与设计人员合作,或者开发人员不是特别精通 JavaScript,在想方设法逐步熟悉移动 Web 开发,那么这种方法就很吸引人。

#### Sencha Touch 适合你吗?

较之与 jQuery Mobile 和 jQTouch, Sencha Touch 学起来难度相对比较在,不过有弊也利:虽然比较复杂,但是带来了强大的功能。作为 Sencha 公司的一款双许可证(商业版和 GPL/FLOSS 版)产品, Sencha Touch 采用了与上述的几种移动开发框架全然不同的方法,原因在于布局和界面窗口组件是使用出色的 JavaScript 库构建而成的,而该 JavaScript 库恰好拥有丰富的实用特性,比如离线支持、独特布局和轻松制作主题的功能。

此外, Sencha Touch 采取了从极其全面的角度来应对应用程序开发所固有的挑战,原因在于它支持可以完全直接开发 MVC 驱动的应用程序。为了形象地说明开发 Sencha Touch 应用程序与开发 jQuery Mobile / jQTouch 应用程序之间的反差到底有多明显,不妨将 jQuery Mobile / jQTouch 演示程序里面的源代码与用于仅仅创建和启动视口的下列 Sencha Touch 代码作一比较:

```
Ext.regApplication({ name: 'App',
  defaultUrl: 'Index/index',
  launch: function()
  { this.viewport = new App.views.Viewport();
  },
});
```

虽然这种方法对相对不熟悉 JavaScript 的新

人来说可能难度很大,不过我还是建议花点时间尝试一下 Sencha Touch,因为直观的语法、编写清晰的文档以及众多的配套实例对于你尽快上手会有大有帮助。

#### Sencha Touch 的主要特性

虽然使用 Sencha Touch 的理由有多个,但是下面三个理由显得尤为突出:

原生的 MVC 架构致力于为面向框架的开发人员消除紧张,他们可能在为如何合理地组织和管理 JavaScript 代码而苦恼。Sencha 开发人员已付出了很大的努力,在 Sencha 网站上发布了众多的教程和演示应用程序。

对于之前熟悉窗口组件工具包(想一想 Swing 和 GTK+)的开发人员来说,窗口组件的布局和组织方面采用 JavaScript 驱动的方法自然很熟悉。

#### 结束语

读者最后总是问我应该选择哪一种框架。回答总是一样:这得看情况。毫无疑问, Sencha Touch 是这种开发框架当中功能最丰富的,不过有时候学起来比较困难。话虽如此,如果你的目的仅仅是为现有的应用程序提供对用户友好的移动界面,那么 jQuery Mobile 或 jQTouch 可能会提供一条阻力最小的道路。如果你给移动应用程序设的目标比较高,那么 Sencha Touch 极有可能完全值得你为此投入时间。不管你最后选择了什么样的解决方案,我都很想听听你使用后的感受! 欢迎留言。■

#### 原文地址

<http://mobile.51cto.com/web-321296.htm>

## 如何把交易型元数据变成移动应用？

也许一些人会觉得很意外,但我们能从总部设在波士顿的新兴公司 SCVNGR 获益良多;确切地说,能从其移动支付平台 LevelUp 获益良多。这家公司正在从头开始重新考虑交易的性质,奉行一个独特的原则:交易就是数据。

货币不就是某种有价值的数据的实体表现吗? 有价值的数据经交易后多次易手,而交易本身带来了众多数据,传达了大量信息:购买了什么商品、谁购买、哪里购买以及如何购买。我们可以称之为交易型元数据。LevelUp 现在有足够的钱通过其生态系统不断流入,它获得交易型元数据后,可以将它变成可编程的应用程序。正因为如此,该公司在上周发布了一套应用编程接口(API),那样开发人员就能在交易的基础上开发应用程序了。如果你认真想一想,会发觉这是个非常值得关注的现象。

### 交易型元数据的价值

LevelUp 是一款应用程序,让你可以用智能手机在实体商家处完成支付。进入到赛百味或奎兹诺斯三明治连锁店,点一份三明治,让收银员扫描你手机上的二维码,就完成了支付。SCVNGR 创始人塞思·普瑞巴什把这比作是另一种信用卡,只不过这种信用卡正好放在你的智能手机中。

普瑞巴什说:“我们开始看到的一幕是,一些比较大的品牌在使用 LevelUp,大量交易开始通过它获得。在一些情况下,我们开始看到通过 LevelUp 进行的交易比通过美国运通 (America Express) 进行的交易还要多;由于交易数量庞大,我们正在逼近万士达。”

作为一种应用程序,LevelUp 对你以往购物情况的了解比任何普通的信用卡或借记卡都深入得多。

创建这种类型的数据有着巨大的价值,比如你在哪里购买了商品,何时购买,具体购买了哪些商品,经常隔多久去商家处购物,附近有哪些商家。确实,如果支付处理公司想一想“智能”支付这场革命,这正是他们所觊觎的那种数据。移动支付公司 Square 同样径直将目光投向了这种数据,为广大商家推出了“Card Case”活动和新的分析工具。谷歌钱包和 Google Offers 服务都在试图为交易型数据增添价值,像万士达、维萨和美国运通等各大支付处理公司也在如法炮制。支付行业发生的根本性转变正在让交易拥有丰富的可编程数据,然后在此基础上创建服务。

我们谈论的是哪种类型的应用? 你恐怕已经看到过早期版本的应用。已存在了数十年的忠诚度计划可以利用交易型数据,根据购买频率和是否邻近商家,提供有针对性的优惠券。保单和收据可以成为独立的应用程序,或者通过某种类似 Dropbox 的集成功能,集成到现有的应用程序当中。如果交易型数据被纳入进来,可以通过 Foursquare 等现有服务,可以大大提高客户进行评价的便利。

## 如何把交易型元数据变成移动应用？ II

智能支付以及把交易型元数据变成应用程序的想法不是某种考虑不周的梦想。它正在变为现实。比如说,只要看一下 Greylock 合伙公司的 Reid Hoffman、Ali Rosenthal 和 James Slavet 在《福布斯》杂志上写的一篇特约博文:“这里的平台是支付网络。软件开发人员将为智能卡添加新功能,只需为支付网络编程,以便将在线应用程序与特定的支付活动联系起来。消费者将来就能够有效地把应用程序‘拖放’到智能卡中,就像今天他们把应用程序添加到智能手机中那样。”

### 硬币的两个面

我们可以一直谈论支付方法的价值,但是利用交易型元数据创造价值涉及的面比如何购买产品涉及的面还要多。它可能是 LevelUp 那样的二维码、谷歌钱包那样的 NFC、PayPal 或 Square 那样的数字钱包,具体方法其实并不重要。这每一种服务的目的是,创建可以用来构建网络的数据。不过,这些数据又是如何传送的?交易需要一些途径将这些数据发送到合适的云,以便它经过处理后,能够以服务、优惠券或应用程序的方式发回去,那样它才能为消费者带来价值。在这种情况下,销售点系统变得同等重要。

NFC 和谷歌钱包项目面临的困难之一是,眼下支持它们的基础架构很有限。万士达的 PayPass 终端可以处理 NFC,但是需要升级软件,才能处理优惠券。事实上,虽然支付技术在不断进步,但是作为交易型元数据的中心,背后的 POS 系统却落后一步。Square 正在借助其 Register iPad 应用程序竭力改变这种局面。Square 弄清楚了支付行业的未来,推出了一款自上而下的集成

解决方案。它从交易方法入手,它在 Register 中收集信息,然后传送给 Card Case。整条环路就算完成。Square 的下一步是打算扩大这条环路,添加更多的价值。LevelUp 已逐渐明白了 POS 系统同样至关重要。它在上周发布的一套 API 让 POS 集成商和制造商可以添加 LevelUp,作为万士通卡、维萨卡和美国运通卡等支付平台之外的一个独特选项。普瑞巴什说:“等它集成到销售点系统后,我们可以开始为商家提供分析工具,不仅可以分析销售数量和客户,还能分析具体购买了哪些商品。”LevelUp 在直接追逐 POS 运营商,从商家手里接过了集成的重任。Square 势必需要将其 Register iPad 应用程序交到广大零售商的手里,而 LevelUp 可以直接预先装在 POS 供应商的系统中,有望迅速实现规模化。LevelUp 应该会在今后几周宣布与几家知名 POS 供应商结为合作伙伴的消息。普瑞巴什说:“这些销售点系统公司说‘嗨,我们大大小小的客户要求这个支付平台,所以我们会在支付节点为 LevelUp 搭建一个平台。所以,我们支持万士通卡,我们支持维萨卡,支持发现卡和 LevelUp,我们肯定会添加支持该支付平台的功能。如果你能提供平台级支持,这给众多零售场所带来了机会,它们同样可以使用 LevelUp。所以说,这确实让人很兴奋。’”

LevelUp 目前每个月处理的交易金额为 100 万美元,这个数字在其发展过程中显得很重要。不过,如果 LevelUp 想与各大公司一争高下,还是有许多工作需要跟上。Square 现在每年处理的交易金额多达 40 亿美元,支付处理公司发现通过这些移动支付平台流动的资金高达数万亿美元。■

# 不破不立？移动行业的创造性破坏

一边是激增的移动互联网流量——归因于更先进的客户端设备，另一边是高涨的基础设施投资——投入到改进的 3G 网络和功能强大的 4G 网络，整个行业在尝试开发新的服务以及新的盈利模式。

运行更多种操作系统的新设备旨在带来一批成本低廉的智能手机，这类智能手机的目标群体是仍然为数众多的没有在使用 BlackBerry、iPhone 或 Android 手机的人。

一边是激增的移动互联网流量——归因于更先进的客户端设备，另一边是高涨的基础设施投资——投入到改进的 3G 网络和功能强大的 4G 网络，整个行业在尝试开发新的服务以及新的盈利模式。加特纳德国公司的研究副总裁 Bettina Tratz-Ryan 说：“眼下，移动运营商从服务方面实际获得的移动数据收入非常少，大部分的收入还是来自实行固定收费的数据服务订购。所以，为了证明部署 LTE 的合理性——部署 LTE 需要比 3G 网络更多的蜂窝小区，以便增加带宽，移动运营商必须开拓客户体验更出色的服务。比如说，那些服务可以从质量和灵活性等方面来界定，可以与面向消费者的社交媒体进行融合，或者与面向企业用户的统一通信进行融合。”

## Windows Phone 劫后余生？

在设备端，这次盛会上最大的新闻就是微软大刀阔斧地重新设计了 WindowsPhone7 移动操作系统。演示版本仅仅展示了新的用户界面，但它以一种易于浏览和导航的方式，巧妙地融合了版面式样和极简图标设计；应用程序、内容和信

息都汇集在“中心”（hub）里面，这些中心有着共同的组织和导航主题。

微软回避了关于内核、新的开发工具、Silverlight 支持、使用哪种浏览器或者被认为是该“平台”一部分的其他任何组件的所有问题。

不过，用户界面给观众们留下了深刻的印象。技术咨询公司 CurrentAnalysis 的分析师 AviGreengart 在博文中评述这则新闻时写道：“这是消费者一直所期待的彻底变化，为了能够再度使用微软产品。WindowsPhone7 系列全面具有竞争力——娱乐、企业级应用和个人办公套件，无论如何。”微软并没有直接谈到面向企业用户的变化或改进，但是演示版本显示了面向微软 Office 套件的“Office 中心”（OfficeHub），包括 OneNote 记事应用程序，以及访问 SharePoint 这一微软的企业协作、工作流和文档管理系统的重要功能。不过仍有人对这款操作系统能不能与苹果、RIM 以及越来越强大的 Android 一较高下表示出了怀疑。资深的微软观察人士 Joe Wilcox 在一篇博文中写道，WindowsPhone7 “出师未捷身先死”。他认为，微软已失去了太多的市场份额和心灵占有率，又面临来自苹果和谷歌 Android 这些太多成功竞争者的压力，其移动操作系统很难杀出重围。



## 不破不立？移动行业的创造性破坏 II

### 运营商应对变化

设备领域的创新突显了移动运营商们在一个迅速变化的行业继续面临的困境。谷歌首席执行官 Eric Schmidt 在 MWC 上对观众说，谷歌的软件工程师们现在首先关注移动平台方面的开发，其次才关注桌面方面的开发。这种转变背后的原因恰恰反映了移动行业出现的根本性变化。

据 iSuppli 公司的董事兼首席分析师 Jagdish Rebello 声称，谷歌在积极大胆地拓展移动搜索服务和应用程序，并将它们与定位服务和移动广告服务联系起来，它有望按照自己的设想重塑移动行业：服务免费提供给最终用户，用户以点击广告的方式来付费。

Rebello 写道：“与移动价值链的其余环节一样，谷歌在不遗余力地发掘新的用户行为模式，并通过推广云存储和云计算、移动广告以及各种基于位置的服务，推动社交网络服务的发展。谷歌的所有免费服务都朝着这个目标而努力。”

在巴塞罗那，移动运营商们披露了最近应对变化的尝试。24 家运营商联合起来，发布了一款统一的移动应用开发平台，以便与苹果的应用程序商店和谷歌的 Android 市场一较高下。成立大规模应用社区（WAC）的目的是为移动软件开发人员提供一种跨平台的开放编程标准，旨在让他们更容易、更迅速地将开发的应用程序投放市场。

但是多种操作系统和多家应用程序商店仍必须得到支持。澳大利亚最大的移动运营商澳洲电讯（Telstra）计划创建一个在线“购物中心”，用户们可以随意浏览其中的商店，选择专门针对其手机的应用程序。澳洲电讯的首席技术官

Hugh Bradlow 在 MWC 展会上说：“我们将打造一种购物中心环境，这意味着我们是价值链中绕不开的一个环节。”澳洲电讯目前不是大规模应用社区（WAC）的成员。

### LTE 继续前进

移动行业向基于长期演进（LTE）标准的新型移动基础设施的步伐一直没有停下来：

就基于 LTE 的语音（voice-over-LTE）规范达成了协议，这一方面是为了实现 LTE 服务的标准化，并且确保 LTE 设备可以在不同的网络上运行。

中国移动运营商 CSL 已在香港完成了商业 LTE 试用的第一个阶段，原型 USB 调制解调器的下载速率高达 100Mbps。

日本的 NTT DoCoMo 展示了一款内置 LTE 调制解调器的原型笔记本电脑。

AT&T 最近宣布为扩建其 LTE 网络选择了两家基站厂商。

但是分析师们表示，这些资本投入并非最终目的。关键还是服务，通常与互联网公司进行合作。英国市场研究公司 Informa Telecoms & Media 的首席分析师 Thomas Wehmeier 在一篇博文中指出，Facebook 的高管们在 MWC 上介绍了他们与英国移动运营商沃达丰英国公司的联合实验项目。这两家公司在尝试让目前未在使用数据服务的沃达丰用户可以免费访问 Facebook。Wehmeier 写道：“两家公司取得了巨大的成功，在试用精简版 Facebook 服务的那些用户当中有 20% 将数据‘扩充服务’添加到了每月的服务方案中。” ■

## ■ 编者按

虽然高效、正确地编程对于确保 Android 应用程序顺畅运行、迅即响应大有帮助,但是有些应用程序得益于在 Android 的 Java 虚拟机(名为 Dalvik VM)边界的外面运行代码。这些应用程序而是以原生方式运行代码。

## Android编程方法大PK: NDK vs. RenderScript

Android 平台为应用程序在传统的 Android 应用边界外面运行提供了两种方法。第一种、也是应用最广泛的方法是使用原生开发工具包(NDK)。第二种方法是使用 RenderScript(RS),这是一种低级的高性能编程语言。这两种机制都适用于 3D 渲染和处理器密集型计算。

我们在本文中将比较 NDK 和 RS,帮助你决定何时使用其中一种方法。

### NDK vs. RenderScript: 编程语言和可移植性

NDK 让开发人员可以用 C 或 C++ 编程,并通过 Java 原生接口 (JNI) 机制与 Android 应用程序进行联系。可用的库是标准库,基本上不需要变更,就可以常常移植现有的 C/C++ 代码。此外, C++ 与 Java 区别不是很大,许多开发人员同时精通这两种语言。

RenderScript 则采取了不同的方法,它使用 C99 语法(标准 C 来自 1999 年,最新标准是 C11,来自 2011 年),新的应用编程接口 (API) 最终编译成原生代码。虽然这种语法广为人知,但是使用这套系统面临一个学习过程,因为其 API 并非广为人知。

最终的可移植性体现在有没有在种类最广泛的设备和平台上使用的库。NDK 让你可以利用现有的 C/C++ 库,你可能已经在其他平台上使用这些库。RenderScript 无法从其他 C 应用程序移植过来,不过它在 Android 设备上比 NDK

更为常见。比如说,RenderScript 可在谷歌电视 (Google TV) 设备上运行,但是目前 NDK 还缺少对谷歌电视平台的支持,即使它运行 Android。

### NDK vs. RenderScript: 编译和调试

用 NDK 编写的代码必须事先针对每一个目标原生平台来编译。如果应用程序在架构未得到支持的平台上运行,应用程序的 NDK 代码部分就无法正常运行。RenderScript 在你的开发机器上进行第一遍编译,然后在目标设备上进行最后一遍编译,因而带来了更高效的原生二进制代码。这意味着,凡是支持 RenderScript 的设备都可以运行你的代码,不管采用什么架构。

目前,RenderScript 带来的代码只能在主处理器上运行,它会自动生成可以利用多个核心的代码——如果目标设备上有多个核心。不过在将来,有计划让 RenderScript 代码在图形处理器 (GPU) 上也可以运行。这类似 CUDA 或 OpenCL 平台。



## Android 编程方法大 PK: NDK vs. RenderScriptII

采用 NDK 的应用程序可以使用 gdb 进行行级调试。另一方面, RenderScript 应用程序在运行时无法调试。考虑到 RenderScript 具有的性质及其处理多个核心的方式,这没什么好大惊小怪的,不过这也加大了查找和消除代码错误的难度。



### NDK vs. RenderScript: 性能

NDK 和 RenderScript 都未能在性能方面提供完美方案。两者都增加了项目的复杂性,降低了可移植性,提高了测试需求,加大了调试难度,还给项目增加了维护负担。如果你的项目不需要进行大量计算,只使用 OpenGL 的基本图像功能,或者已经在足够快速地运行,那么 NDK 和 RenderScript 都不太可能给项目带来足够明显的好处。

如果纯粹是用于计算, RenderScript 的设置和配置很容易,最终的运行速度实际上可能胜过使用 NDK 的类似实现方法,需要编写的代码比较少。RenderScript 最适合处理 3D 用户界面或高性能计算任务。另一方面, NDK 比较适合高性能 OpenGL 应用程序或需要访问图形软件开发

工具包 (SDK) 更多功能或访问第三方库的游戏。

简单的 OpenGL 任务或不受制于处理器的计算任务最好别去管它。Java 编译器和 Dalvik VM 的性能总是在不断提升。就让你的代码继续使用 Java,这让你编写的应用程序可以充分利用这些性能上的提升,在将来的 SDK 版本或设备上可以更好地运行。

随着最后一个编译步骤得到改进,为 GPU 添加更多的硬件支持和计算支持, RenderScript 代码在将来可能会有所改进。另一方面,除了通过硬件改动获得的性能提升外,通过 NDK 编写的本地代码不太可能出现性能提升。因此, NDK 代码从高效的算法和代码得到的好处最大。

### 结束语

最后,选择使用 NDK、RenderScript 还是继续使用 Java,完全取决于开发人员。应用程序设计方面的这个决定具有重大影响:它影响着你使用什么编程语言、编写的应用程序可以在什么设备上运行,以及从维护的角度来看你的软件项目有多复杂。

你已经了解了 NDK 和 RenderScript 的诸多优缺点。它们未必可以换着使用,但在许多情况下,可以用这两种技术开发出相似的解决方案。了解 NDK 和 RenderScript 的工作机理,可以帮助你作出更明智的决定,决定在具体开展某个项目时使用哪一种方法。不管怎样,目前有工具可以帮助你让自己编写的应用程序在尽可能多的设备上尽可能快速地运行。■

### 原文地址

<http://mobile.51cto.com/hot-321085.htm>

## ■ 编者按

经常在魅族社区混的网友可能都知道美女开发者“野百合”，她开发的“换发型”到“摇传”、“心情壁纸”等多款应用因为简单实用而获得了不少的关注，并一度成为魅族人气开发者 No.1。

# 美女开发者独立开发二十多款应用，月入上万

这位个人开发者非计算机科班出身，靠着自学学习 Android 开发，至今已经开发了二十多款应用，其中“心情壁纸”还是一款收费应用。据“野百合”透露，目前她所开发的应用通过广告联盟等途径，已经为她挣得了大约每月 1 万左右的收入。对于众多仍挣扎在应用开发盈利线上的个人开发者来说，“野百合”的故事可以说非常励志。注：采访中嘉宾提到了对一些具体公司的褒贬，仅代表嘉宾本人意见，51CTO 仅原样呈现采访实录，不表示我们同意或否认嘉宾观点。



(图：@无拘无束的野百合)

51CTO：请简单介绍一下你自己（年龄段，职业背景，与应用开发有关的经历，目前的职业等）。

野百合：我是个宅女，勉强算是八零后，本身学的专业不是计算机，毕业之后在深圳工作过几年，受所处环境的影响，开始接触一些专业的东西，以前做的都是和手机应用没有关联的工作。

51CTO：你所有的应用都是一个人做的吗？从构思到设计到开发，平均做一个应用需要花多少时间？时间最短的和最久的分别是哪一个？

野百合：基本上是吧，技术难题会请教高人指点。做应用的时间还真是有长有短，最短的应该是一个“计时器”，花了一晚上时间，最长大概是“平衡积木”。

51CTO：第一款应用是什么，什么时间做的？

野百合：第一款应用是“违章查询”，去年三月底上架的。

51CTO：目前为止一共做了多少款应用了，其中收费的几款？

野百合：一共大概二十几个，收费的只有“心情壁纸”一款。

51CTO：从“换发型”到“摇传”到“电子书DIY”，虽然都是工具类，但这些应用差别还挺大的，每次你是如何决定下一个应用的方向？想到什么做什么？还是有计划的？

野百合：其实我完全没有计划着做什么应用，都是自己需要什么就做什么，比如说我开车的时候，在高速上被拍了好几次照，因为不知不觉就超速了，所以我做了“GPS 出行助手”，让它一到 110 公里就提醒我。要么就是自己觉得什么好玩就做什么，就比如说我特别喜欢玩大家来找茬的游戏，躺床上能玩好几个小时，所以我就自己做了好多好多关卡，把它们拼成了“终级找茬”。

51CTO：应用开发方面，你觉得你的特长是什么，短板又是什么？

野百合：真不好意思，我还真是没什么特长，这可不是谦虚，但也没什么短板，呵呵。



## 美女开发者独立开发二十多款应用,月入上万 II

51CTO: 你开发的应用都是首先通过什么渠道发布? 除了魅族和应用汇还发布在哪些地方?

野百合: 跟所有普通的开发者一样, 我没有资金去做推广, 唯一的渠道就是在各应用市场上架, 所以应用市场有什么要求, 我们就乖乖地听话。

51CTO: 很多人对国内用户的付费习惯有质疑, 所以包括一些大公司在内, 他们都会考虑免费提供应用然后通过广告或增值服务的方式收费, 这种环境下你为什么“敢”收费?

野百合: 的确, 除了国内用户不适应付费以外, 还有就是付费渠道尚不完善, 小的开发者大多都只是加个广告条什么的来勉强糊口。

我做收费其实也只是个尝试, 因为魅族的收费平台已经做得很好了, 也有众多的铁杆煤油。说到魅族, 我自己有一部 M9, 我觉得这是国产机里最漂亮的, 去年我为它做了一些应用, 到去年夏天的时候, 一度成为魅族开发者第一名。

再后来, 因为有一些新的广告形式, 比如要使用积分, 魅族市场不接受这种形式, 所以很多应用都没有再上架了, 挺遗憾的。

51CTO: 能否透露一下目前收费应用的总收入(大概的量级), 和总用户数? 其中收入最多的是哪一款? 用户数最多的?

野百合: “心情壁纸”收入不多, 一两千块钱, 和其他应用的广告收入相比, 有点不值一提, 其他应用的广告收入每个月大概有一万块左右。但这是我自己很喜欢的一个应用, 制作它的那些日子里我很用心。

51CTO: 除了应用收费, 是不是还通过其他

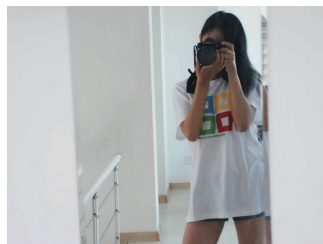
方式盈利, 比如广告联盟什么的?

野百合: 对, 就像前边所说的, 主要的收入还是靠植入广告。这里想多说几句, 和普通的开发者息息相关的就是广告平台, 国内广告平台目前有几百家, 每一家的市场人员都在通过各种方式联络开发者, 希望占领更多的应用。我觉得, 开发者看重的第一是技术, 第二是服务。

一个广告平台可以提供多少种形式, 让开发者选择这很重要, 我觉得万普做得很好, 他的很多形式在刚推出的时候都会受到排斥, 但最终都成为别人模仿的对象。

当然, 要说服务, 也是万普做得最好, 有哪家的 CEO 一天 24 小时 QQ 在线充当客服呢? 黄波会在群里跟大家承认有哪些小 bug, 会向大家收集改进意见, 这种态度很难能可贵。

再说到打款, 有的平台后半个月才可以申请提前半个月的钱, 而后半个月的提款申请, 到下个月才会执行, 这种小聪明开发者都可看得出来的。



(2011 年 7 月 15 日, 野百合成为魅族人气开发者 NO.1 穿着魅族赠的 T 恤留影)

51CTO: 应用发布之后都通过什么方式推广?

野百合: 有些开发者可能会在论坛发帖, 或者有一些其他方式吧, 但是我还没有试过。

## 美女开发者独立开发二十多款应用,月入上万 III



(开发这些应用野百合只用了半年的时间)

51CTO: “心情壁纸”添加了自定义功能后就较少更新了,引起一些用户的抱怨,看来大家对收费应用的后续服务还是比较在意,你对应用“售后服务”这件事怎么看?随着发布的应用越来越多,如果所有应用都继续升级、提供后续服务势必会占据你很大的精力,打算怎么处理这种状况?

野百合:对,这就是我暂时不打算再做收费软件的原因,收费之后,必然是要提供更多的服务,让用户觉得花费值得,但作为个人开发者,特别是像我这样随性而做的人来说,不太现实。

51CTO:一款应用要让所有人都喜欢太难了,现在还会为用户的差评烦恼吗?一般都怎么处理用户反馈?

野百合:我觉得安卓用户对成千上万的安卓

应用是很有包容性的,就算他们不喜欢,留下了差评,也说明至少他们关注过你的应用。所以针对软件的差评真的没什么,就当是一种不动听的鼓励好了。但是如果差评是因为应用里有广告,我会觉得很无奈,有时候看到比较大的电子媒体里登一篇文章,给大家讲如何屏蔽广告,我只能觉得很心寒。

51CTO:近期有什么新的开发计划?

野百合:希望往苹果开发转移,然后就是好好想一想,怎么样让我的那些不懂编程的朋友也能在移动互联的大潮里分一杯羹。

51CTO:对于未来,你有什么样的规划和梦想?

野百合:我是个女人,没有雄心大志,就想像现在这样自由安逸地生活,做更多自己喜欢的应用。

51CTO:你的网名是“无拘无束的野百合”,野百合有什么特质让你特别喜欢?

野百合:因为我很喜欢那首老歌“野百合也有春天”,呵呵

51CTO:除了做手机应用,业余有什么爱好?

野百合:喜欢做美食,喜欢看美剧,喜欢做手工,喜欢弹吉它,还喜欢看到社会不公平现象在微博上发发牢骚。

51CTO:如果让你用一句话阐述对生活的感受,你会说什么?

野百合:我会说:因为自由,才会活得精彩。

### 原文地址

<http://mobile.51cto.com/team-322145.htm>

## ■ 编者按

从发布前历次都会出现的样品流出,苹果的饥渴营销策略总能吊足了大家的胃口,甚至有些腻。但这次,苹果此次发布的新 iPad 令欢乐我感到失望。

## 杂谈：迷失的苹果新款iPad

之前的苹果产品拥有一个独有的精神——改变世界,至少其产品总能让人真的眼前一亮,而这次发布的名叫“新 iPad”的平板产品,却走了大多手机厂商走的硬件更新的老路,苹果已经迷失了么?



新款 iPad 和老款有什么不同?

我本寄希望于苹果发布一款更睿智的产品来改变我对创新的认知。可惜,这次我终于高估苹果了。当然,苹果新 iPad 发布会发布了些“新”东西的,欢乐我就带大家再回顾并品评一下苹果“新 iPad”的新特性:

### 1、名字

此次 iPad 并无单独名称,官方叫法即为“The new iPad”(全新的 iPad)。据悉,在苹果之后的 iPad 发布中也将沿袭这一做法,只发布“新的 iPad”,而不再单独命名。

欢乐评论:如同欢乐上面串词的多格漫画,苹果新 iPad 的名称之离奇,令人乍舌。欢乐我英文水平太烂实在搞不懂这里的深意,就中文产品型号名字来说,至少我活这么大还真没有见哪个商家这么用的。这样的命名不难让我猜想,假如这款产品我使用了 2 年后已经磨损的残破不堪的时候,别人问我:你 iPad 是什么型号? 我将如何解释……——尴尬

### 2、屏幕

屏幕方面,苹果新推出的 iPad 采用 Retina 屏幕,分辨率为 2048 × 1536,是上一代产品的 4 倍。

欢乐评论:曾经让已故的乔布斯赞许有加的 Retina 屏幕终于用到了其平板产品上,至于 Retina 屏的特点,看过 iPhone4 或 4S 的同学早已知晓,没看过的小编简单的告诉大家就是画面表现特别细腻、特别高清、特别养眼、特别……但这个特性毕竟是我们早已了解的,只是苹果刚刚把它用在平板上,手法早已被各大媒体想到,概念更不算新了。——失望

### 3、摄像头

新款 iPad 的摄像头较上代产品有所改进,此前 iPad 2 的后置摄像头的像素是 70 万,现在则达到 500 万。用户可以借此拍摄 1080p 的高清视频短片。

欢乐评论:500 万像素真的很了不起,但想想前几天诺基亚 4100 万的像素,似乎逊色了不少。

## 杂谈：迷失的苹果新款 iPad II

姑且不论其他镜头参数,单就苹果你自己宣扬的 500 万已经被竞争对手的 4100 万甩的老远。诺基亚在推出 4100 万手机后,不知道是不是因此原因,股价大跌。不知苹果对自己的 500 万摄像头的宣传卖点还有多少信心。至少目前看来,手机或平板的摄像头完全取代传统相机还需时日,500 万摄像头也不过如此。——麻木

### 4、处理器

新款 iPad 搭载 A5X 处理器,上代 iPad 则是 A5 处理器。苹果表示, A5X 处理器速度是英伟达 Tegra 3 的两倍,性能是后者 4 倍。

欢乐评论:“核战”开始。N 核处理器的概念最早出现在 PC 机上,后来衍伸到笔记本领域,现在又来到了手机领域,我觉得战况和结果也许可以参考传统的处理器大战:那就是“核”的更新速度远远高于软件发展速度,远远多于人们日常的使用需求。现在的消费类处理器市场人们早已不在只追求运算速度,适合自己的性价比高的产品才能胜出。现在手机领域的“核战”也以五花八门,速度早已不是绝对卖点。就拿苹果来说, iPhone 4S 比 4 快多少,比 3GS 又快多少,拿现在流行的软件大多很难比较的出来,软硬如何结合,这才是关键。——还是麻木

### 5、网络

新一代 iPad 支持 4G LTE 网络,美国市场 iPad 将分为 AT&T 和 Verizon 两个版本。

欢乐评论:在中国,4G 的概念过于超前,还是先解决一下 3G 的问题比较务实。等 4G 真的落地我国后,我们再来感受下苹果这超前的模块设计。——尚难体会

### 6、iOS 5.1/Siri

新款 iPad 支持最新 iOS 5.1 系统,依照苹果的系统命名习惯,可以看出本次的 5.1 系统应该不会有过多的改变。令人不解的是,新款 iPad 竟然未能支持 Siri 语音助理! 苹果把类 Siri 功能的语音输入开关加入到新 iPad 的核心升级中,可以用于语音输入。

欢乐评论:如果说 4S 与 iPhone4 最大的差别就在于对 Siri 的支持的话,那么新款 iPad 显得弱爆了! 其实 4S 真的和 4 没有什么差别,大多数人还是冲着 Siri 这个新特性购买的 4S。库克你给大家个理由买你的新 iPad? ——弱爆了!

### 7、其他方面

新款 iPad 厚 9.4 毫米,超过二代的 8.8 毫米;重量为 635 克,超过二代的 601 克。

欢乐评论:重量增加必然使得轻便性下降,虽然幅度不大。当年二代的超薄概念令人眼前一亮,这次新 iPad 的超重概念也让人眼前一亮。——死胖纸!

### 欢乐小结:

苹果的产品难道已经逐步沦为平庸? 乔布斯的离去使得苹果产品的水准尽失? 至少这次苹果没有交出令人满意的答案。也许乔布斯还在冥冥中保护着这个万众瞩目的企业,但长此以往,光环终将褪去。

总结了这些,大家肯定以为我是个“果黑”,其实对其更多的批评才能显出我对苹果这个企业的热爱。苹果曾经改变了世界,我喜欢的就是它这曾经拥有的精神。但这次的新款 iPad 丢了这股精气神,在我眼里,它真的已经迷失了……■



51CTO技术沙龙 第18期

# 大话PHP设计模式



2012年3月4日举办的第18期51CTO技术沙龙上,51CTO有幸请到了51CTO技术总监韩旭和逗游技术经理/架构师韩友洪两位老师,两位老师分别从思想和实战两个方面来与大家分享自己多年的开发经验心得。

第一位出场的是韩旭老师,韩旭老师曾经任职新浪多年,现任51CTO技术总监,韩旭老师的主题是《程序员经验杂谈》,该主题比较杂,范围比较广,但韩旭老师从一个大家都比较熟悉的美国经典大片《猩球崛起》说起,将枯燥的理论变得有趣生动。

接下来,是韩友洪老师带来的《PHP设计模式》,韩老师的演讲更多的偏向实战。

想设计自己的PHP框架吗?想自己打造团队的基础软件库吗?如果回答“是”,那就需要掌握设计模式。本次议题将简述设计模式相关历史、应用场景,并简述几个典型模式在框架设计中的应用。

专题链接: <http://developer.51cto.com/developer/51cto-salon-18/>



51CTO 技术总监 韩旭

<http://new.51cto.com/exp/salon/speaker.html>

韩旭老师曾经任职新浪多年,现任51CTO技术总监



逗游技术经理 韩友洪

<http://new.51cto.com/exp/salon/speaker.html>

6年PHP、4年JAVA开发经验,也关注nodejs, scala, python等相关语言和敏捷软件过程。曾在新浪参与贴吧、论坛、邮箱的维护、架构设计和开发工作。